# Field 718 SOC (revisions)

Jon Wells

April 20, 2020

## Contents

# 1 Packages and Data

Packages are set and data brought into R:

```
#Load required packages
library(FME)
library(ggplot2)
library(reshape2)
library(SoilR)
library(knitr)
library(rmarkdown)
library(kableExtra)
library(xtable)
library(dplyr)
library(tidyr)
library(nlme)
library(lme4)
library(lmerTest)
library(emmeans)
library(multcomp)
library(multcompView)
library(openxlsx)

#install.packages("devtools")
#library(devtools)
# Install SoilR from GitHub. This is the version with the new code. If you install from CRAN you won't be a
#install_github("MPIBGC-TEE/SoilR-exp/pkg")

#document version and sessioninfo
Sys.info()
```

```
##           sysname           release           version          nodename
##         "Windows"          "10 x64"     "build 18362" "CROW-BIOCHAR-PC"
##           machine             login              user    effective_user
##          "x86-64"       "Jon Wells"       "Jon Wells"       "Jon Wells"
```

```
sessionInfo()
```

```
## R version 3.6.3 (2020-02-29)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 18362)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] openxlsx_4.1.4     multcompView_0.1-8 multcomp_1.4-12    TH.data_1.0-10
##  [5] MASS_7.3-51.5      survival_3.1-8     mvtnorm_1.1-0      emmeans_1.4.5
##  [9] lmerTest_3.1-1     lme4_1.1-21        Matrix_1.2-18      nlme_3.1-144
## [13] tidyr_1.0.2        dplyr_0.8.4        xtable_1.8-4       kableExtra_1.1.0
```

```
## [17] rmarkdown_2.1      knitr_1.28         SoilR_1.2.103      reshape2_1.4.3
## [21] ggplot2_3.2.1      FME_1.3.6.1        coda_0.19-3        rootSolve_1.8.2
## [25] deSolve_1.27.1
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.3          lattice_0.20-38     zoo_1.8-7
##  [4] assertthat_0.2.1    digest_0.6.25       R6_2.4.1
##  [7] plyr_1.8.5          evaluate_0.14       httr_1.4.1
## [10] pillar_1.4.3        rlang_0.4.5         lazyeval_0.2.2
## [13] rstudioapi_0.11     minqa_1.2.4         nloptr_1.2.2
## [16] splines_3.6.3       sets_1.0-18         webshot_0.5.2
## [19] readr_1.3.1         stringr_1.4.0       igraph_1.2.4.2
## [22] munsell_0.5.0       compiler_3.6.3      numDeriv_2016.8-1.1
## [25] xfun_0.12           pkgconfig_2.0.3     htmltools_0.4.0
## [28] tidyselect_1.0.0    tibble_2.1.3        expm_0.999-4
## [31] codetools_0.2-16    viridisLite_0.3.0   crayon_1.3.4
## [34] withr_2.1.2         grid_3.6.3          gtable_0.3.0
## [37] lifecycle_0.1.0     magrittr_1.5        scales_1.1.0
## [40] zip_2.0.4           estimability_1.3    stringi_1.4.6
## [43] xml2_1.2.2          vctrs_0.2.3         boot_1.3-24
## [46] sandwich_2.5-1      tools_3.6.3         glue_1.3.1
## [49] purrr_0.3.3         hms_0.5.3           parallel_3.6.3
## [52] yaml_2.2.1          colorspace_1.4-1    rvest_0.3.5
## [55] minpack.lm_1.2-1
```

```r
#read in data, set baseline/years
fractions <- read.xlsx(paste0(mainDir, dataDir, "/718_FractionData_Used in book chapter.xlsx"), sheet = "fo
```

# 2 Table and figure statistics

linear mixed models (LMM) were used to compute basic stats on the presented data collected across the 4 year study (baseline - year 4).

Below we bring in and label the data:

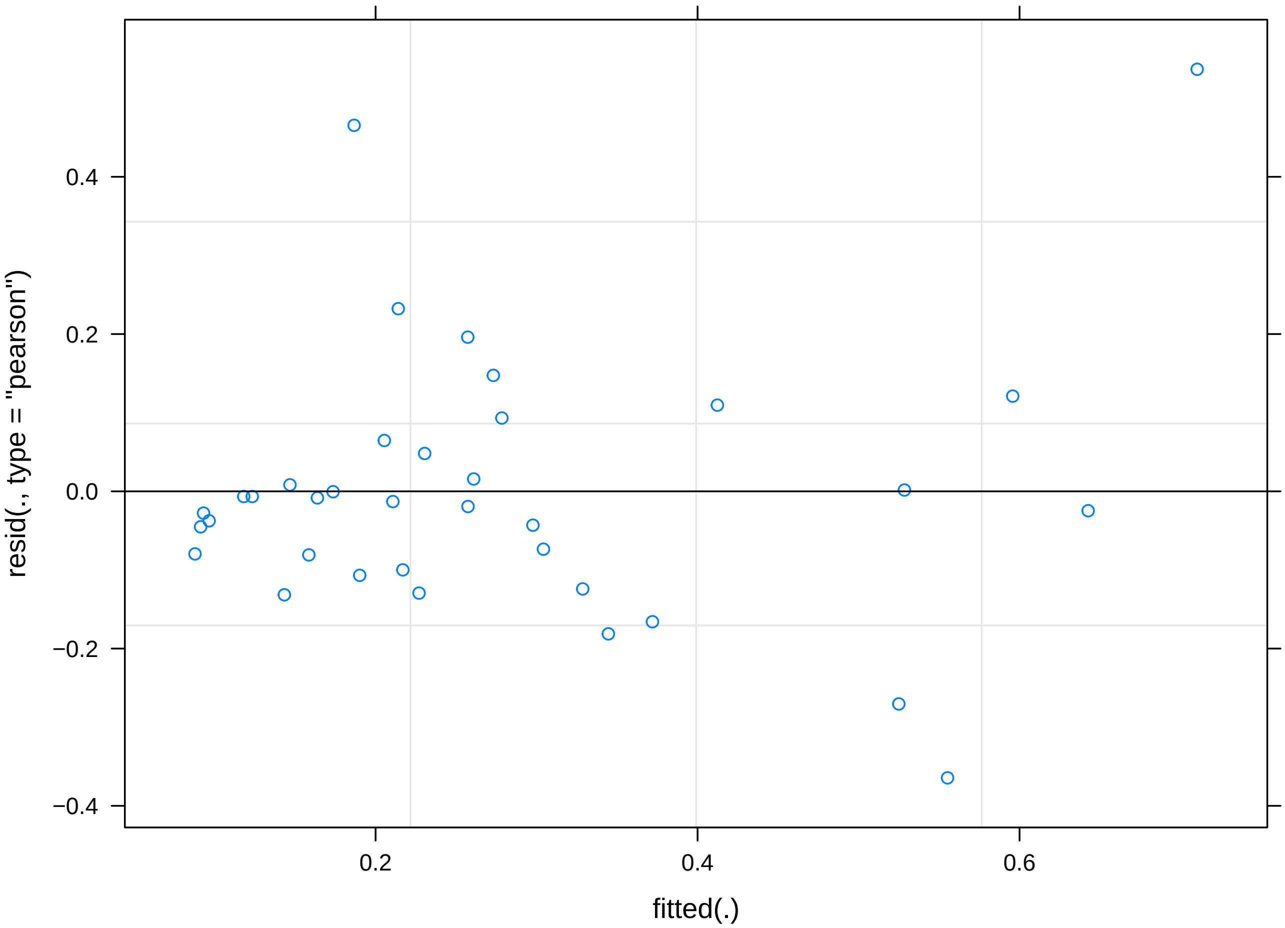
```
#redefine main directory for stats in revisions folder
statsDir <- "C:/Users/Jon Wells/OneDrive/Dissertation/Papers/718 Soil Fractions/Revisions"

#read in root data
root <- read.xlsx(paste0(statsDir,"/2020.02.13 GCB combined data.xlsx"),
                  sheet = "Roots for R")
colnames(root)[c(1:6)] <- c("crop","year","rep","depth", "c","MgC.ha")
root.t <- aggregate(MgC.ha ~ year + rep, data=root, FUN=sum)
#read in aboveground data
agb <- read.xlsx(paste0(statsDir,"/2020.02.13 GCB combined data.xlsx"),
                 sheet = "Raw AGB data")
#subset excel sheet to aboveground yield and C dataframes
agy <- agb[c(1:12), c(1:2,4:5,9)]
colnames(agy)[c(1:5)] <- c("crop","year","rep","P-R","MgC.ha.yr")
agc <- agb[c(1:6), c(13:15,17)]
colnames(agc)[c(1:4)] <- c("year","rep","agc.perc","bgc.perc")
#read in ESM from surface and deep soils
soc <- read.xlsx(paste0(statsDir,"/2020.02.13 GCB combined data.xlsx"),
                 sheet = "ESM for R")
colnames(soc)[1:5] <- c("crop", "rep", "year", "ESM.3600", "ESM.18000")
#read in fraction data
socf <- read.xlsx(paste0(statsDir,"/2020.02.13 GCB combined data.xlsx"),
                  sheet = "Fractions for R")
socf <- socf[c(1:15),c(1:3,8:10)]
```

## 2.1 Roots

Root data were collected by depth on the first and second year of the trial (not during baseline). The root data was evaluated using a Linear mixed model (LMM) with depth as a fixed effect and rep nested within year as a random effect. Nesting Rep within year accounts variance between reps and how that variance changes with time (here only two time points).

In response to the use of repeated measures ANOVA, LMM can accomplish everything a repeated measures ANOVA can (and much more) with more flexibility and robustness. Repeated measures ANOVAs will never be better or more accurate than mixed modeling.

```
#factor depth and year for models
root$depth <- as.factor(root$depth)
root$year  <- as.factor(root$year)
#Roots - linear mixed model of root carbon by depth w/ time as random effect
r.mg.c <- lmer(MgC.ha ~ depth + (1|rep/year), data=root)
op <- par()
par(mfrow=c(2,2))
plot(r.mg.c)
```




```r
par(op)
summary(r.mg.c)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: MgC.ha ~ depth + (1 | rep/year)
##    Data: root
##
## REML criterion at convergence: 0.9
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.89035 -0.44401 -0.08393  0.27114  2.78433
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  year:rep (Intercept) 0.003109 0.05576
##  rep      (Intercept) 0.006514 0.08071
##  Residual             0.037163 0.19278
## Number of obs: 36, groups:  year:rep, 6; rep, 3
##
## Fixed effects:
##              Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)   0.59291    0.09425  9.87437   6.291 9.53e-05 ***
## depth100-120 -0.43719    0.11130 25.00011  -3.928 0.000596 ***
```

```
## depth20-40   -0.29802    0.11130 25.00011  -2.678 0.012910 *
## depth40-60   -0.33836    0.11130 25.00011  -3.040 0.005483 **
## depth60-80   -0.38168    0.11130 25.00011  -3.429 0.002107 **
## depth80-100  -0.43189    0.11130 25.00011  -3.880 0.000673 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) d100-1 d20-40 d40-60 d60-80
## dpth100-120 -0.590
## depth20-40  -0.590  0.500
## depth40-60  -0.590  0.500  0.500
## depth60-80  -0.590  0.500  0.500  0.500
## depth80-100 -0.590  0.500  0.500  0.500  0.500
```

```r
#contrasts by depth
cld.root.d <- cld(emmeans(r.mg.c, ~ depth),
                alpha = 0.05,
                Letters = letters,
                ordered= TRUE, reversed=TRUE)
kable(cld.root.d, "latex", booktabs=TRUE) %>%
  kable_styling(c("striped"))
```

| | depth | emmean | SE | df | lower.CL | upper.CL | .group |
|---|---|---|---|---|---|---|---|
| 2 | 0-20 | 0.5929052 | 0.0942526 | 9.874339 | 0.3825345 | 0.8032759 | a |
| 6 | 20-40 | 0.2948890 | 0.0942526 | 9.874339 | 0.0845183 | 0.5052596 | ab |
| 5 | 40-60 | 0.2545484 | 0.0942526 | 9.874339 | 0.0441778 | 0.4649191 | ab |
| 4 | 60-80 | 0.2112231 | 0.0942526 | 9.874339 | 0.0008525 | 0.4215938 | b |
| 3 | 80-100 | 0.1610121 | 0.0942526 | 9.874339 | -0.0493586 | 0.3713828 | b |
| 1 | 100-120 | 0.1557190 | 0.0942526 | 9.874339 | -0.0546517 | 0.3660896 | b |

## 2.2 Contrasts presented in Table 1

Several things to clear up for the reviewer: 1. Applying a repeated measures ANOVA to an experiment where time is the only treatment is simply an ANOVA. 2. ANOVAs and linear models (LM) are the same thing. 3. Linear mixed models (LMM) are better and more flexible (e.g. you can do things like account for rep variation)

So, everything presented in the paper is accurate and defensible. However, Dr. Kantar suggests replacing the GLS analysis with a LMM to make the paper more parsimonious. I previously tested rep as a random effect in LMMs and it has no effect on the outcome of any of the simple linear models. Considering we already did a linear model (which is an ANOVA, and in the case of our data where time is the only treatment is a repeated measures ANOVA) we don't really need LMMs, but the gist of Mikey's suggestions were to use LMMs cause they're better and use LMMs for everything for parsimony.

### 2.2.1 Above ground yield

Above ground biomass yield (AGY) was evaluated using a LMM with year as a fixed effect and rep as a random effect. To deomnstrate that in this case the AGY results are the same for ANOVA, LM, and a LMM, all three analyses were completed below:

```r
#AGY - anova of agy
agy$year <- as.factor(agy$year)

#ANOVA
lm.agy <- aov(MgC.ha.yr ~ year, data=agy)
op <- par()
par(mfrow=c(2,2))
```

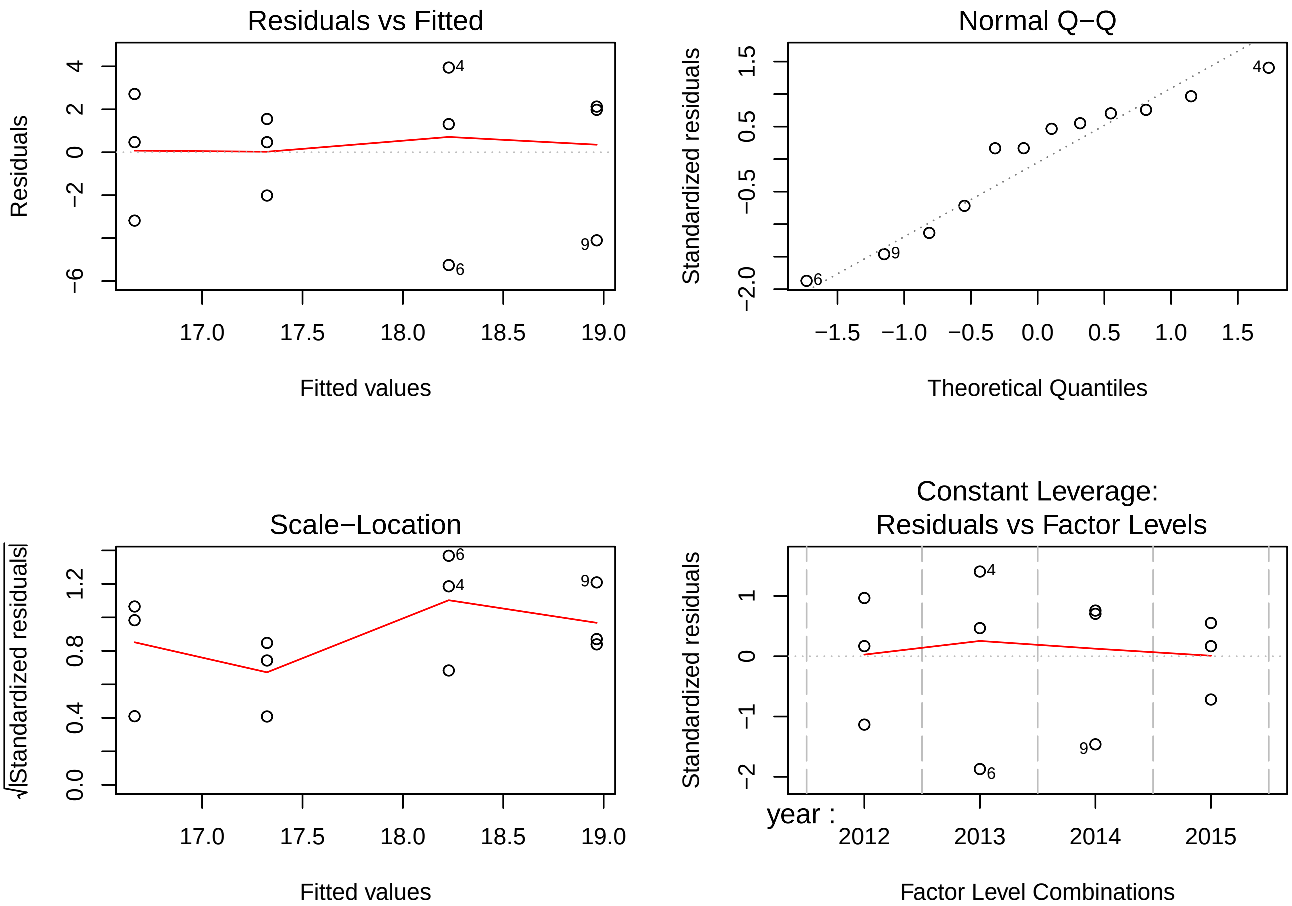
```r
plot(lm.agy)
par(op)
summary(lm.agy)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## year         3   9.18   3.061   0.259  0.853
## Residuals    8  94.53  11.817
```

```r
#lsmeans test and compact letter display
cld.agy <- cld(emmeans(lm.agy, ~ year),
             alpha = 0.05,
             Letters = letters)
cld.agy1 <- cld.agy[order(cld.agy$year),]
#save and show table of results
kable(cld.agy1,"latex", booktabs=TRUE) %>%
  kable_styling(c("striped"))
```

| year | emmean | SE | df | lower.CL | upper.CL | .group |
|---|---|---|---|---|---|---|
| 2012 | 16.66340 | 1.984678 | 8 | 12.08672 | 21.24008 | a |
| 2013 | 18.22843 | 1.984678 | 8 | 13.65176 | 22.80511 | a |
| 2014 | 18.96551 | 1.984678 | 8 | 14.38884 | 23.54219 | a |
| 2015 | 17.32307 | 1.984678 | 8 | 12.74639 | 21.89975 | a |

```r
#Linear model (LM)
lm.agy <- lm(MgC.ha.yr ~ year, data=agy)
op <- par()
par(mfrow=c(2,2))
plot(lm.agy)
```



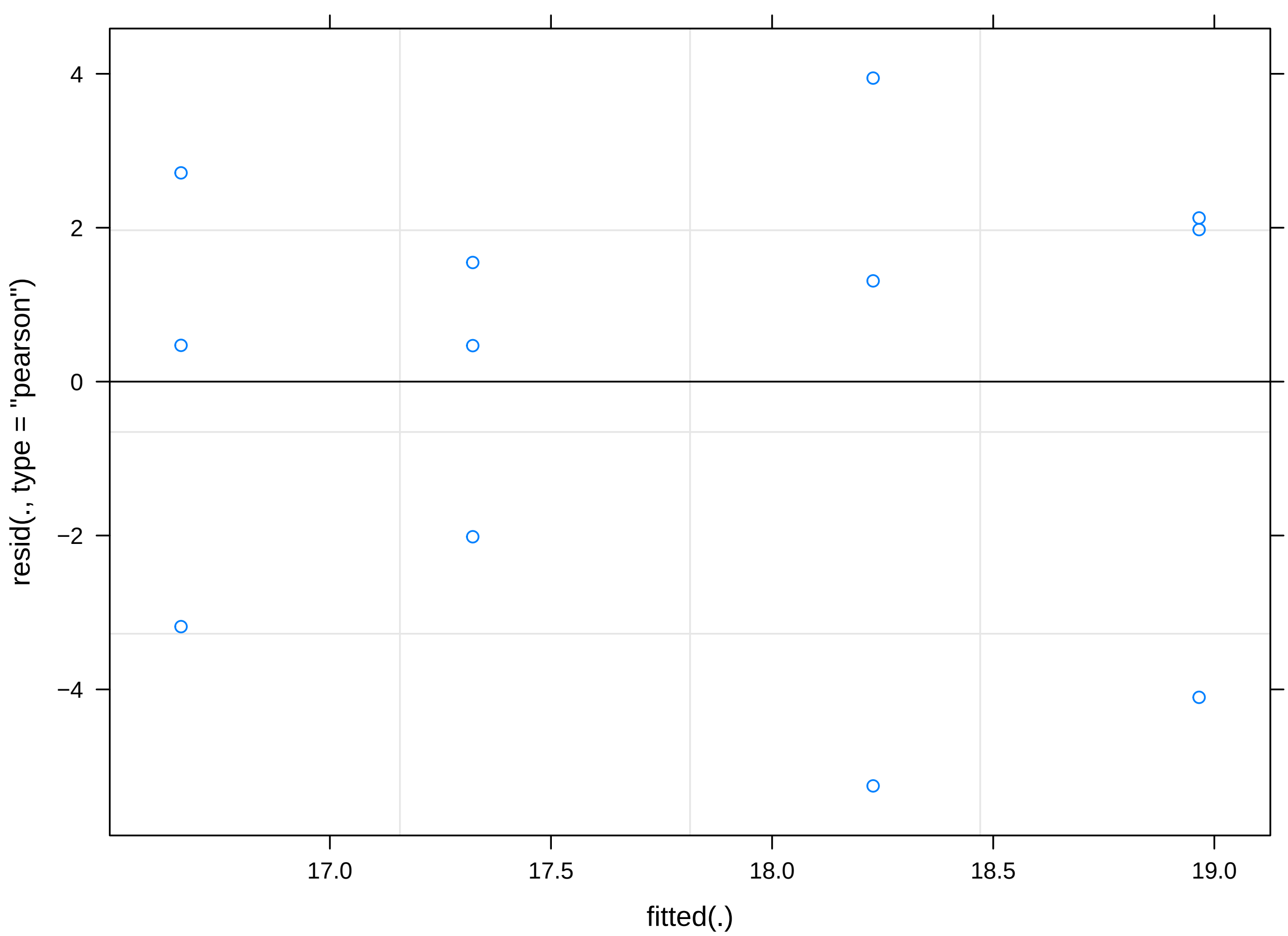

```r
par(op)
summary(lm.agy)
```

```
##
## Call:
## lm(formula = MgC.ha.yr ~ year, data = agy)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.2538 -2.3080  0.8902  2.0135  3.9446
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  16.6634     1.9847   8.396 3.08e-05 ***
## year2013      1.5650     2.8068   0.558    0.592
## year2014      2.3021     2.8068   0.820    0.436
## year2015      0.6597     2.8068   0.235    0.820
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.438 on 8 degrees of freedom
## Multiple R-squared:  0.08854,    Adjusted R-squared:  -0.2533
## F-statistic: 0.2591 on 3 and 8 DF,  p-value: 0.8529
```

```r
#lsmeans test and compact letter display
cld.agy <- cld(emmeans(lm.agy, ~ year),
               alpha = 0.05,
               Letters = letters)
cld.agy2 <- cld.agy[order(cld.agy$year),]
#save and show table of results
kable(cld.agy2,"latex", booktabs=TRUE) %>%
  kable_styling(c("striped"))
```

| year | emmean | SE | df | lower.CL | upper.CL | .group |
|---|---|---|---|---|---|---|
| 2012 | 16.66340 | 1.984678 | 8 | 12.08672 | 21.24008 | a |
| 2013 | 18.22843 | 1.984678 | 8 | 13.65176 | 22.80511 | a |
| 2014 | 18.96551 | 1.984678 | 8 | 14.38884 | 23.54219 | a |
| 2015 | 17.32307 | 1.984678 | 8 | 12.74639 | 21.89975 | a |

```r
#Linear mixed model (LMM)
lm.agy <- lmer(MgC.ha.yr ~ year + (1|rep), data=agy)
op <- par()
par(mfrow=c(2,2))
plot(lm.agy)
```




```r
par(op)
summary(lm.agy)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: MgC.ha.yr ~ year + (1 | rep)
##    Data: agy
##
## REML criterion at convergence: 46.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.5284 -0.6714  0.2590  0.5857  1.1475
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  rep      (Intercept)  0.00    0.000
##  Residual             11.82    3.438
## Number of obs: 12, groups:  rep, 3
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)  16.6634     1.9847  8.0000   8.396 3.08e-05 ***
## year2013      1.5650     2.8068  8.0000   0.558    0.592
## year2014      2.3021     2.8068  8.0000   0.820    0.436
## year2015      0.6597     2.8068  8.0000   0.235    0.820
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr) yr2013 yr2014
## year2013 -0.707
## year2014 -0.707  0.500
## year2015 -0.707  0.500  0.500
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

```r
#lsmeans test and compact letter display
cld.agy <- cld(emmeans(lm.agy, ~ year),
             alpha = 0.05,
             Letters = letters)
cld.agy3 <- cld.agy[order(cld.agy$year),]
#save and show table of results
kable(cld.agy3,"latex", booktabs=TRUE) %>%
  kable_styling(c("striped"))
```

| year | emmean | SE | df | lower.CL | upper.CL | .group |
|---|---|---|---|---|---|---|
| 2012 | 16.66340 | 1.984678 | 8 | 12.08672 | 21.24008 | a |
| 2013 | 18.22843 | 1.984678 | 8 | 13.65176 | 22.80511 | a |
| 2014 | 18.96551 | 1.984678 | 8 | 14.38884 | 23.54219 | a |
| 2015 | 17.32307 | 1.984678 | 8 | 12.74639 | 21.89975 | a |

### 2.2.2 Surface/Deep SOC

Analyzing SOC for statistical differences across the 4 year trials has a few complications. Most importantly, the baseline of the field was taken using 10 pneumatically drills cores across the entire field, while the cores taken in each rep are done by hand in triplicate. This means that the sample n is different between baseline and all other years and that rep 1-3 from baseline are not equivalent to the reps 1-3 sampled each year afterwards. To overcome this I used a GLS model to account for the difference in variance between years as a random effect.

After consulting Dr. Kantar (Mikey), he recommended that gls was ok but that a LMM would be better though the results are nearly the same for the surface SOC and slightly better parsed for deeper SOC (hence the suggestion for LMM over GLS). He suggets that using LMM would also make the paper's statistical approaches more parsimonious.

In this case the LMM is needed b/c there are large differences in variance between baseline and years 1-4 due to the difference in sampling and sample n; to demonstrate an ANOVA, LM, and LMM are shown below for both surface and deep carbon.

```r
#SOC - had to account for difference in sample n and sd between baseline and years 1-4
#Accomplished using gls and variance differences specified by years
soc$year <- as.factor(soc$year)

#Surface carbon - ANOVA
lm.soc <- aov(ESM.3600 ~ year, data=soc)
op <- par()
par(mfrow=c(2,2))
plot(lm.soc)
par(op)
summary(lm.soc)
```

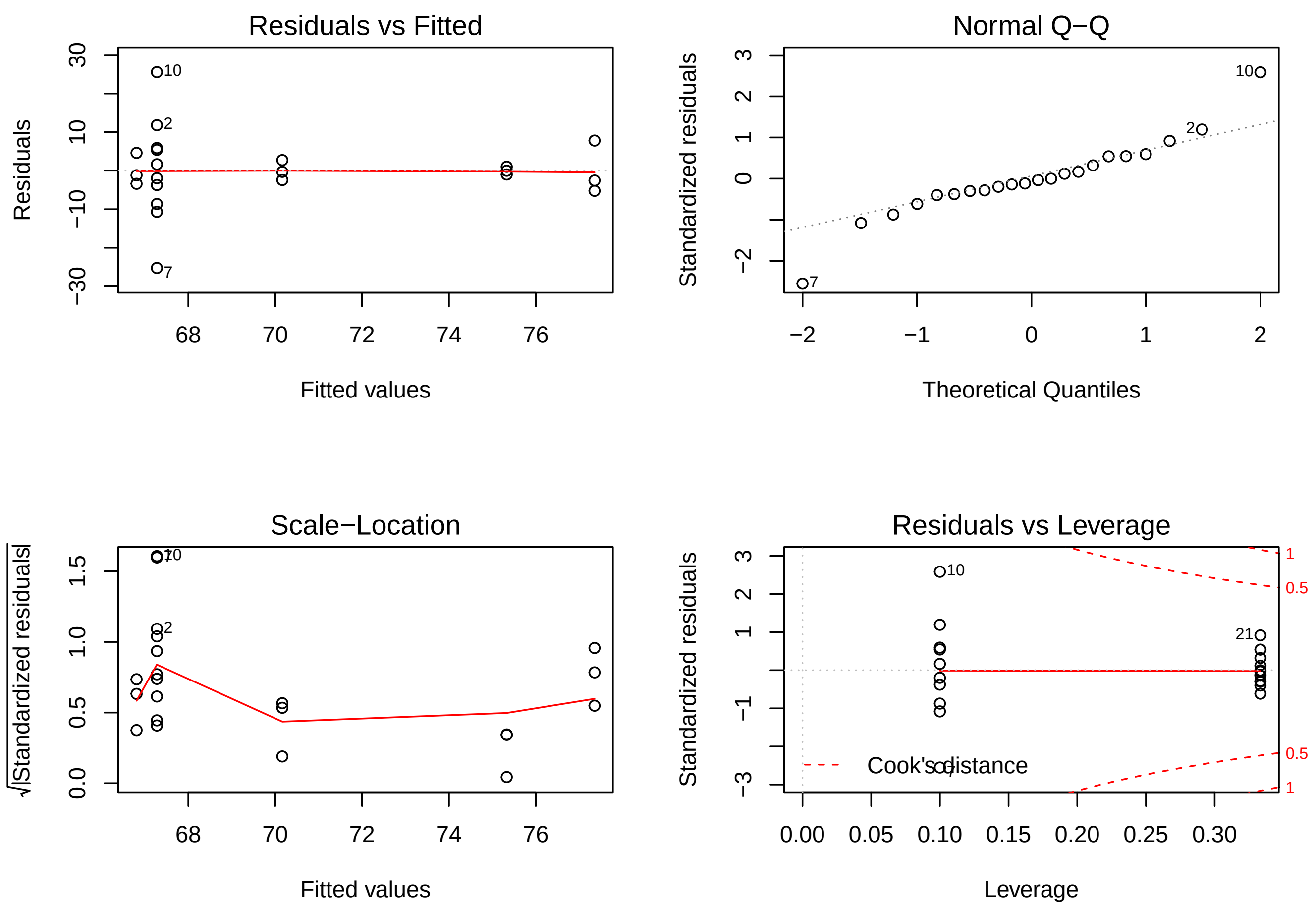
#### 2.2.2.1 Surface SOC

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## year         4  352.1   88.02   0.811  0.536
## Residuals   17 1845.9  108.58
```

```r
#lsmeans test and compact letter display
cld.soc <- cld(emmeans(lm.soc, ~ year),
             alpha = 0.05,
             Letters = letters,
             ordered=TRUE,reversed=TRUE)
cld.soc1 <- cld.soc[order(cld.soc$year),]
#save and show table of results
kable(cld.soc1,"latex", booktabs=TRUE) %>%
  kable_styling(c("striped"))
```

| | year | emmean | SE | df | lower.CL | upper.CL | .group |
|---|---|---|---|---|---|---|---|
| 3 | 2011 | 67.27631 | 3.295203 | 17 | 60.32404 | 74.22858 | a |
| 2 | 2012 | 70.16452 | 6.016190 | 17 | 57.47147 | 82.85758 | a |
| 1 | 2013 | 75.33056 | 6.016190 | 17 | 62.63751 | 88.02361 | a |
| 5 | 2014 | 66.80903 | 6.016190 | 17 | 54.11598 | 79.50208 | a |
| 4 | 2015 | 77.35173 | 6.016190 | 17 | 64.65868 | 90.04478 | a |

```r
#Surface carbon - LM
lm.soc <- lm(ESM.3600 ~ year, data=soc)
op <- par()
par(mfrow=c(2,2))
plot(lm.soc)
```



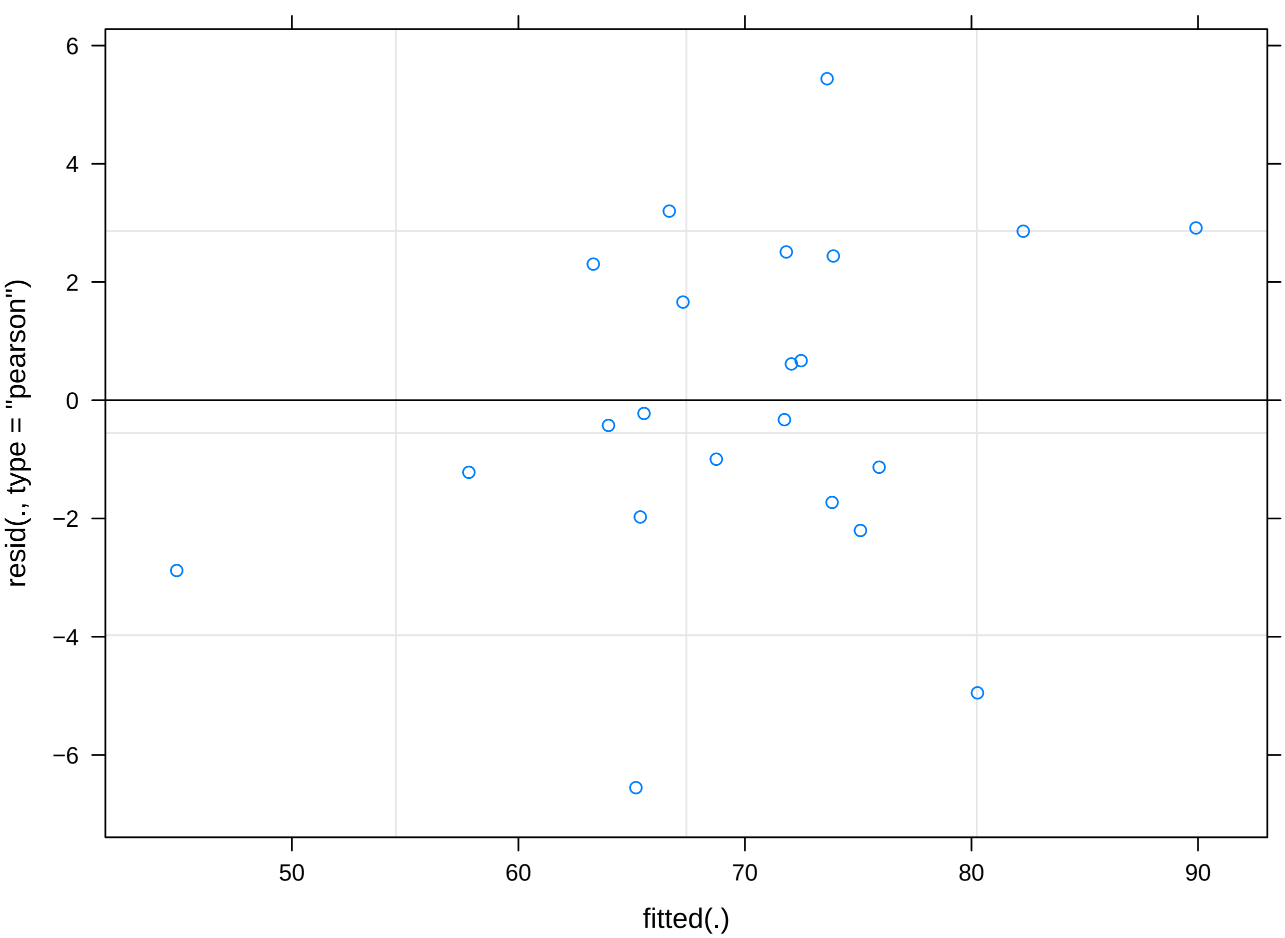

```r
par(op)
summary(lm.soc)
```

```
##
## Call:
## lm(formula = ESM.3600 ~ year, data = soc)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -25.2404  -3.1928  -0.6504   4.1363  25.5488
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  67.2763     3.2952  20.416 2.14e-13 ***
## year2012      2.8882     6.8595   0.421    0.679
## year2013      8.0542     6.8595   1.174    0.257
## year2014     -0.4673     6.8595  -0.068    0.946
## year2015     10.0754     6.8595   1.469    0.160
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.42 on 17 degrees of freedom
## Multiple R-squared:  0.1602, Adjusted R-squared:  -0.03743
## F-statistic: 0.8106 on 4 and 17 DF,  p-value: 0.5355
```

```r
#lsmeans test and compact letter display
cld.soc <- cld(emmeans(lm.soc, ~ year),
               alpha = 0.05,
               Letters = letters,
               ordered=TRUE,reversed=TRUE)
cld.soc2 <- cld.soc[order(cld.soc$year),]
#save and show table of results
kable(cld.soc2,"latex", booktabs=TRUE) %>%
  kable_styling(c("striped"))
```

| | year | emmean | SE | df | lower.CL | upper.CL | .group |
|---|---|---|---|---|---|---|---|
| 3 | 2011 | 67.27631 | 3.295203 | 17 | 60.32404 | 74.22858 | a |
| 2 | 2012 | 70.16452 | 6.016190 | 17 | 57.47147 | 82.85758 | a |
| 1 | 2013 | 75.33056 | 6.016190 | 17 | 62.63751 | 88.02361 | a |
| 5 | 2014 | 66.80903 | 6.016190 | 17 | 54.11598 | 79.50208 | a |
| 4 | 2015 | 77.35173 | 6.016190 | 17 | 64.65868 | 90.04478 | a |

```r
#Surface carbon - LMM with rep as random
lm.soc <- lmer(ESM.3600 ~ year + (1|rep), data=soc)
op <- par()
par(mfrow=c(2,2))
plot(lm.soc)
```



```r
par(op)
summary(lm.soc)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: ESM.3600 ~ year + (1 | rep)
##    Data: soc
##
## REML criterion at convergence: 127.7
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.4986 -0.3660 -0.0631  0.5502  1.2438
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  rep      (Intercept) 148.48   12.185
##  Residual              19.12    4.373
## Number of obs: 22, groups:  rep, 10
##
## Fixed effects:
##             Estimate Std. Error     df t value Pr(>|t|)
## (Intercept)   67.276      4.094  9.035  16.433 4.87e-08 ***
## year2012       1.472      3.499  8.311   0.421   0.6846
## year2013       6.638      3.499  8.311   1.897   0.0930 .
## year2014      -1.883      3.499  8.311  -0.538   0.6045
## year2015       8.659      3.499  8.311   2.475   0.0373 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr) yr2012 yr2013 yr2014
## year2012 -0.134
## year2013 -0.134  0.479
## year2014 -0.134  0.479  0.479
## year2015 -0.134  0.479  0.479  0.479
```

```r
#lsmeans test and compact letter display
cld.soc <- cld(emmeans(lm.soc, ~ year),
             alpha = 0.05,
             Letters = letters,
             ordered=TRUE,reversed=TRUE)
cld.soc3 <- cld.soc[order(cld.soc$year),]
#save and show table of results
kable(cld.soc3,"latex", booktabs=TRUE) %>%
  kable_styling(c("striped"))
```

| | year | emmean | SE | df | lower.CL | upper.CL | .group |
|---|---|---|---|---|---|---|---|
| 3 | 2011 | 67.27631 | 4.093934 | 9.415912 | 58.07717 | 76.47545 | a |
| 2 | 2012 | 68.74860 | 5.063050 | 15.738521 | 58.00090 | 79.49630 | a |
| 1 | 2013 | 73.91463 | 5.063050 | 15.738521 | 63.16693 | 84.66233 | a |
| 5 | 2014 | 65.39310 | 5.063050 | 15.738521 | 54.64541 | 76.14080 | a |
| 4 | 2015 | 75.93580 | 5.063050 | 15.738521 | 65.18810 | 86.68350 | a |

```r
#ANOVA
lm.socd <- aov(ESM.18000 ~ year, data=soc)
op <- par()
par(mfrow=c(2,2))
plot(lm.socd)
par(op)
summary(lm.socd)
```

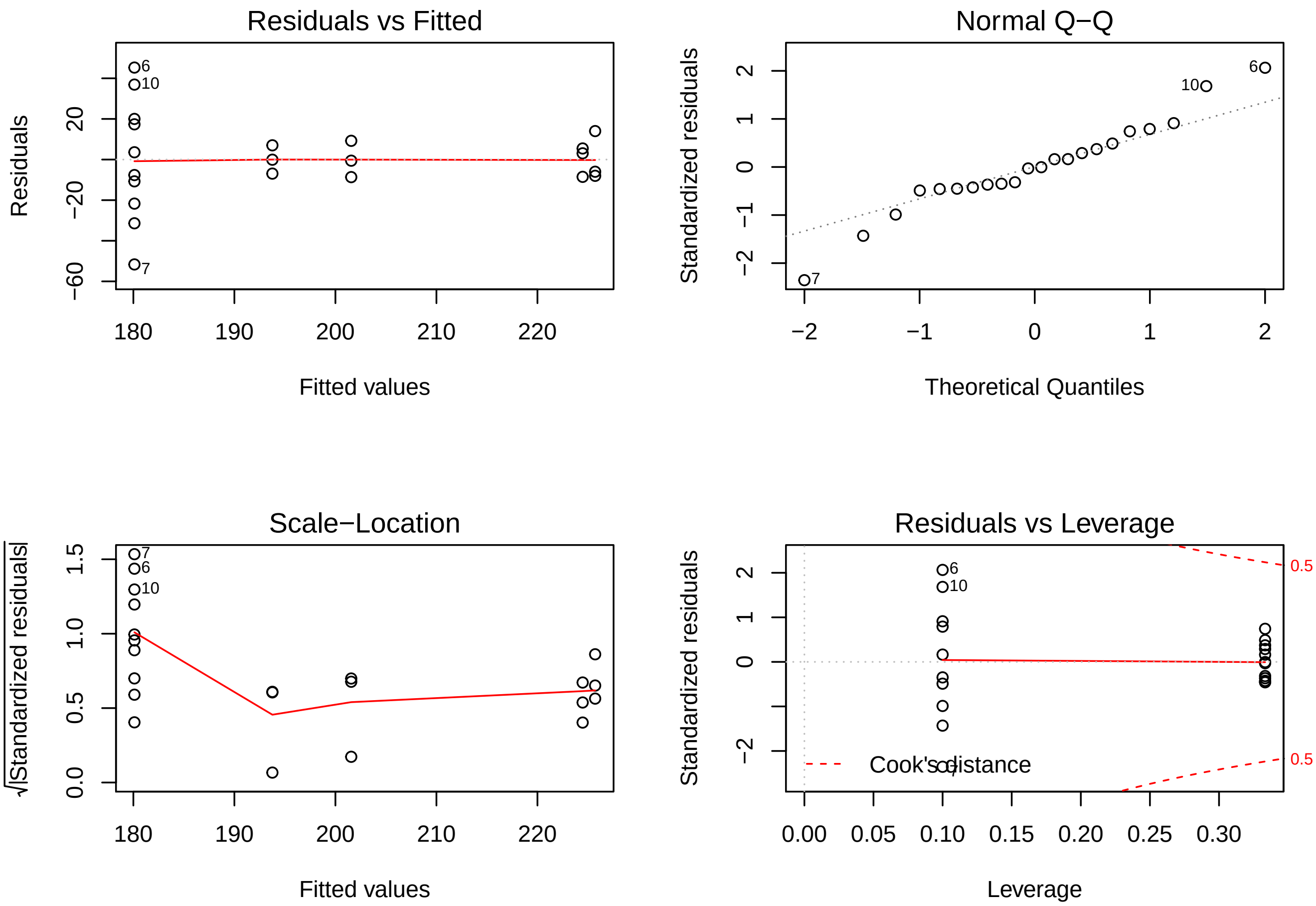
#### 2.2.2.2 Deep SOC

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## year         4   7686  1921.4   3.593 0.0268 *
## Residuals   17   9092   534.8
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
#lsmeans test and compact letter display
cld.socd <- cld(emmeans(lm.socd, ~ year),
             alpha = 0.05,
             Letters = letters,
               ordered=TRUE,reversed=TRUE)
cld.socd1 <- cld.socd[order(cld.socd$year),]
#save and show table of results
kable(cld.socd1,"latex", booktabs=TRUE) %>%
  kable_styling(c("striped"))
```

| | year | emmean | SE | df | lower.CL | upper.CL | .group |
|---|---|---|---|---|---|---|---|
| 5 | 2011 | 180.1039 | 7.313243 | 17 | 164.6743 | 195.5335 | a |
| 3 | 2012 | 193.7611 | 13.352093 | 17 | 165.5906 | 221.9316 | a |
| 2 | 2013 | 224.4747 | 13.352093 | 17 | 196.3043 | 252.6452 | a |
| 4 | 2014 | 201.5666 | 13.352093 | 17 | 173.3962 | 229.7371 | a |
| 1 | 2015 | 225.7105 | 13.352093 | 17 | 197.5400 | 253.8809 | a |

```r
#LM
lm.socd <- lm(ESM.18000 ~ year, data=soc)
op <- par()
par(mfrow=c(2,2))
plot(lm.socd)
```



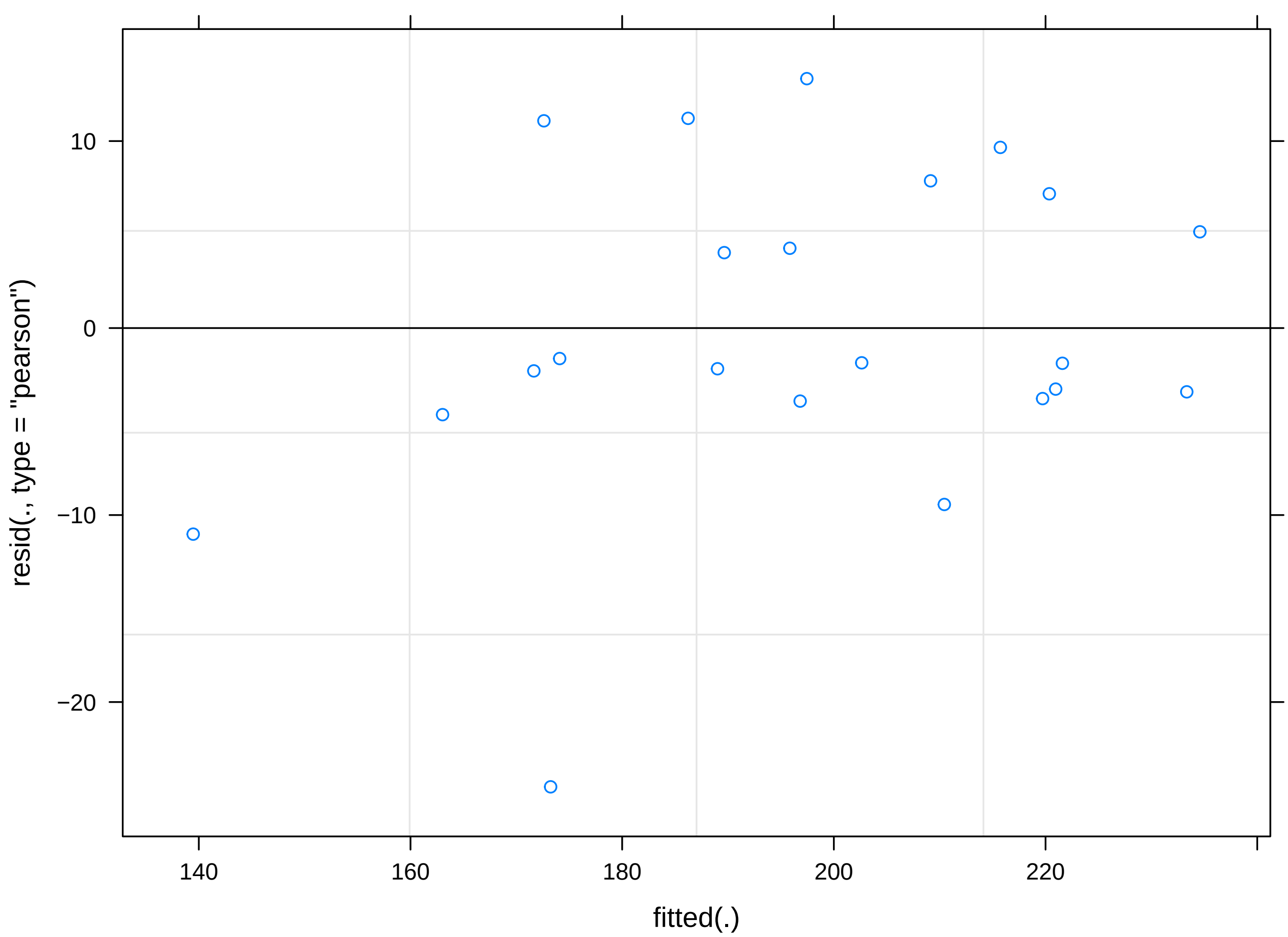

```r
par(op)
summary(lm.socd)
```

```
##
## Call:
## lm(formula = ESM.18000 ~ year, data = soc)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -51.655  -8.397  -0.324   8.669  45.292
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  180.104      7.313  24.627 9.74e-15 ***
## year2012      13.657     15.224   0.897  0.38220
## year2013      44.371     15.224   2.915  0.00966 **
## year2014      21.463     15.224   1.410  0.17663
## year2015      45.607     15.224   2.996  0.00813 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 23.13 on 17 degrees of freedom
## Multiple R-squared:  0.4581, Adjusted R-squared:  0.3306
## F-statistic: 3.593 on 4 and 17 DF,  p-value: 0.0268
```

```r
#lsmeans test and compact letter display
cld.socd <- cld(emmeans(lm.socd, ~ year),
             alpha = 0.05,
             Letters = letters,
               ordered=TRUE,reversed=TRUE)
cld.socd2 <- cld.socd[order(cld.socd$year),]
#save and show table of results
kable(cld.socd2,"latex", booktabs=TRUE) %>%
  kable_styling(c("striped"))
```

| | year | emmean | SE | df | lower.CL | upper.CL | .group |
|---|---|---|---|---|---|---|---|
| 5 | 2011 | 180.1039 | 7.313243 | 17 | 164.6743 | 195.5335 | a |
| 3 | 2012 | 193.7611 | 13.352093 | 17 | 165.5906 | 221.9316 | a |
| 2 | 2013 | 224.4747 | 13.352093 | 17 | 196.3043 | 252.6452 | a |
| 4 | 2014 | 201.5666 | 13.352093 | 17 | 173.3962 | 229.7371 | a |
| 1 | 2015 | 225.7105 | 13.352093 | 17 | 197.5400 | 253.8809 | a |

```r
#LMM
lm.socd <- lmer(ESM.18000 ~ year + (1|rep), data=soc)
op <- par()
par(mfrow=c(2,2))
plot(lm.socd)
```



```r
par(op)
summary(lm.socd)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: ESM.18000 ~ year + (1 | rep)
##    Data: soc
##
## REML criterion at convergence: 158.7
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.9002 -0.2851 -0.1449  0.5169  1.0332
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  rep      (Intercept) 614.7    24.79
##  Residual             166.7    12.91
## Number of obs: 22, groups:  rep, 10
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)  180.104      8.840   8.994  20.374 7.78e-09 ***
## year2012      16.402     10.141   8.504   1.617  0.14221
## year2013      47.116     10.141   8.504   4.646  0.00141 **
## year2014      24.208     10.141   8.504   2.387  0.04229 *
## year2015      48.352     10.141   8.504   4.768  0.00119 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr) yr2012 yr2013 yr2014
## year2012 -0.186
## year2013 -0.186  0.460
## year2014 -0.186  0.460  0.460
## year2015 -0.186  0.460  0.460  0.460
```

```r
#lsmeans test and compact letter display
cld.socd <- cld(emmeans(lm.socd, ~ year),
             alpha = 0.05,
             Letters = letters,
               ordered=TRUE,reversed=TRUE)
cld.socd3 <- cld.socd[order(cld.socd$year),]
#save and show table of results
kable(cld.socd3,"latex", booktabs=TRUE) %>%
  kable_styling(c("striped"))
```

| | year | emmean | SE | df | lower.CL | upper.CL | .group |
|---|---|---|---|---|---|---|---|
| 5 | 2011 | 180.1039 | 8.839692 | 9.877455 | 160.3747 | 199.8332 | b |
| 3 | 2012 | 196.5063 | 12.438078 | 16.937809 | 170.2569 | 222.7557 | ab |
| 2 | 2013 | 227.2199 | 12.438078 | 16.937809 | 200.9705 | 253.4693 | a |
| 4 | 2014 | 204.3118 | 12.438078 | 16.937809 | 178.0624 | 230.5612 | ab |
| 1 | 2015 | 228.4557 | 12.438078 | 16.937809 | 202.2063 | 254.7051 | a |

### 2.2.3 Free light fraction

The free light fraction was evaluated using a LMM with year as a fixed effect and rep as a random effect. As mentioned previously this is a better model than a repeated measures ANOVA, but also as mentioned previously, accounting for the variance of reps has no effect on the model. To deomnstrate an ANOVA, LM, and LMM are shown below.

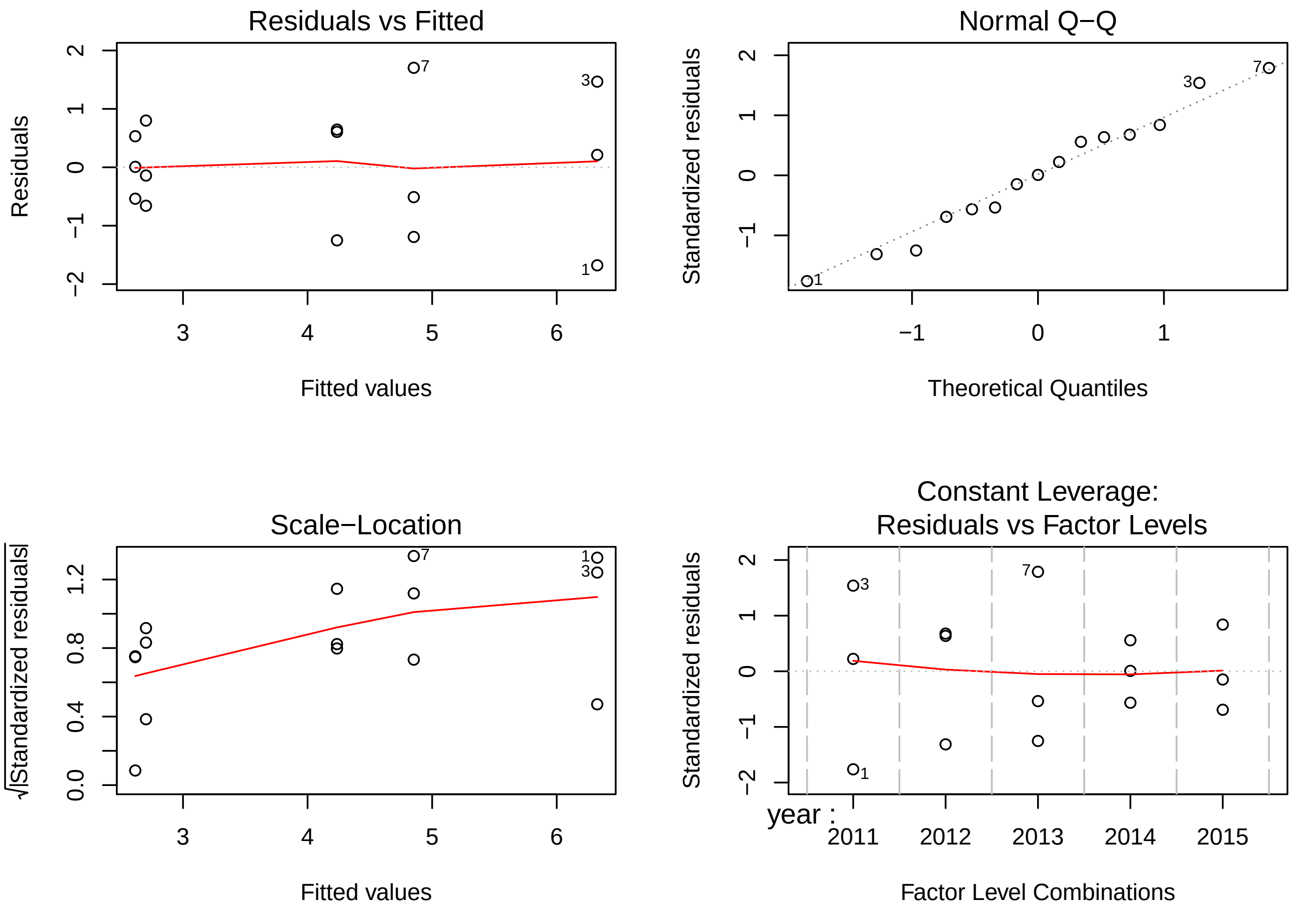
```
#make year catgeorical
socf$year <- as.factor(socf$year)

#ANOVA
lm.free <- aov(fl.c.stock ~ year, data=socf)
op <- par()
par(mfrow=c(2,2))
plot(lm.free)
par(op)
summary(lm.free)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## year         4  29.03   7.257   5.332 0.0146 *
## Residuals   10  13.61   1.361
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#lsmeans test and compact letter display
cld.free <- cld(emmeans(lm.free, ~ year),
                alpha = 0.05,
                Letters = letters,
                ordered=TRUE,reversed=TRUE)
cld.free1 <- cld.free[order(cld.free$year),]
#save and show table of results
kable(cld.free1, "latex", booktabs=TRUE) %>%
  kable_styling(c("striped"))
```

| | year | emmean | SE | df | lower.CL | upper.CL | .group |
|---|---|---|---|---|---|---|---|
| 4 | 2011 | 6.325013 | 0.6735842 | 10 | 4.824174 | 7.825852 | a |
| 2 | 2012 | 4.236032 | 0.6735842 | 10 | 2.735192 | 5.736871 | ab |
| 5 | 2013 | 4.852449 | 0.6735842 | 10 | 3.351609 | 6.353288 | ab |
| 1 | 2014 | 2.616855 | 0.6735842 | 10 | 1.116016 | 4.117695 | b |
| 3 | 2015 | 2.702727 | 0.6735842 | 10 | 1.201888 | 4.203566 | b |

```
#LM
lm.free <- lm(fl.c.stock ~ year, data=socf)
op <- par()
par(mfrow=c(2,2))
plot(lm.free)
```



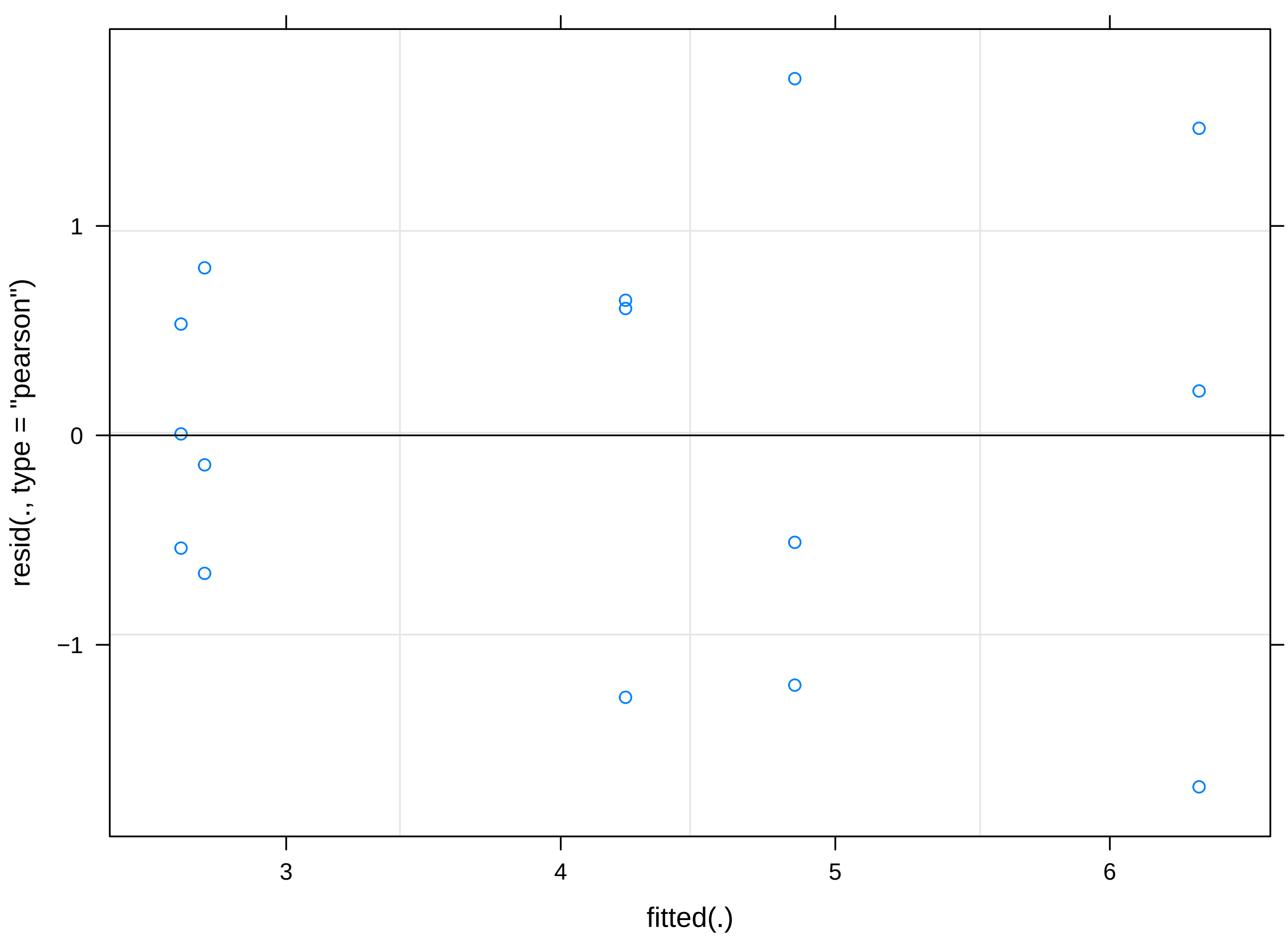

```r
par(op)
summary(lm.free)
```

```
##
## Call:
## lm(formula = fl.c.stock ~ year, data = socf)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.67833 -0.59872  0.00692  0.62545  1.70347
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.3250     0.6736   9.390 2.82e-06 ***
## year2012     -2.0890     0.9526  -2.193  0.05307 .
## year2013     -1.4726     0.9526  -1.546  0.15318
## year2014     -3.7082     0.9526  -3.893  0.00300 **
## year2015     -3.6223     0.9526  -3.803  0.00347 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.167 on 10 degrees of freedom
## Multiple R-squared:  0.6808, Adjusted R-squared:  0.5531
## F-statistic: 5.332 on 4 and 10 DF,  p-value: 0.0146
```

```r
#lsmeans test and compact letter display
cld.free <- cld(emmeans(lm.free, ~ year),
                alpha = 0.05,
                Letters = letters,
                ordered=TRUE,reversed=TRUE)
cld.free2 <- cld.free[order(cld.free$year),]
#save and show table of results
kable(cld.free2, "latex", booktabs=TRUE) %>%
  kable_styling(c("striped"))
```

| | year | emmean | SE | df | lower.CL | upper.CL | .group |
|---|---|---|---|---|---|---|---|
| 4 | 2011 | 6.325013 | 0.6735842 | 10 | 4.824174 | 7.825852 | a |
| 2 | 2012 | 4.236032 | 0.6735842 | 10 | 2.735192 | 5.736871 | ab |
| 5 | 2013 | 4.852449 | 0.6735842 | 10 | 3.351609 | 6.353288 | ab |
| 1 | 2014 | 2.616855 | 0.6735842 | 10 | 1.116016 | 4.117695 | b |
| 3 | 2015 | 2.702727 | 0.6735842 | 10 | 1.201888 | 4.203566 | b |

```r
#LMM
lm.free <- lmer(fl.c.stock ~ year + (1|rep), data=socf)
op <- par()
par(mfrow=c(2,2))
plot(lm.free)
```



```r
par(op)
summary(lm.free)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: fl.c.stock ~ year + (1 | rep)
##    Data: socf
##
## REML criterion at convergence: 37
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.43855 -0.51318  0.00594  0.53609  1.46010
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  rep      (Intercept) 0.000    0.000
##  Residual             1.361    1.167
## Number of obs: 15, groups:  rep, 3
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)   6.3250     0.6736 10.0000   9.390 2.82e-06 ***
## year2012     -2.0890     0.9526 10.0000  -2.193  0.05307 .
## year2013     -1.4726     0.9526 10.0000  -1.546  0.15318
## year2014     -3.7082     0.9526 10.0000  -3.893  0.00300 **
## year2015     -3.6223     0.9526 10.0000  -3.803  0.00347 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr) yr2012 yr2013 yr2014
## year2012 -0.707
## year2013 -0.707  0.500
## year2014 -0.707  0.500  0.500
## year2015 -0.707  0.500  0.500  0.500
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

```r
#lsmeans test and compact letter display
cld.free <- cld(emmeans(lm.free, ~ year),
                alpha = 0.05,
                Letters = letters,
                ordered=TRUE,reversed=TRUE)
cld.free3 <- cld.free[order(cld.free$year),]
#save and show table of results
kable(cld.free3, "latex", booktabs=TRUE) %>%
  kable_styling(c("striped"))
```

| | year | emmean | SE | df | lower.CL | upper.CL | .group |
|---|---|---|---|---|---|---|---|
| 4 | 2011 | 6.325013 | 0.6735842 | 10 | 4.824174 | 7.825852 | a |
| 2 | 2012 | 4.236032 | 0.6735842 | 10 | 2.735192 | 5.736871 | ab |
| 5 | 2013 | 4.852449 | 0.6735842 | 10 | 3.351609 | 6.353288 | ab |
| 1 | 2014 | 2.616855 | 0.6735842 | 10 | 1.116016 | 4.117695 | b |
| 3 | 2015 | 2.702727 | 0.6735842 | 10 | 1.201888 | 4.203566 | b |

### 2.2.4 Occluded light fraction

The occluded light fraction was evaluated using a LMM with year as a fixed effect and rep as a random effect. As mentioned previously this is a better model than a repeated measures ANOVA, but also as mentioned previously, accounting for the variance of reps has no effect on the model. To deomnstrate an ANOVA, LM, and LMM are shown below.

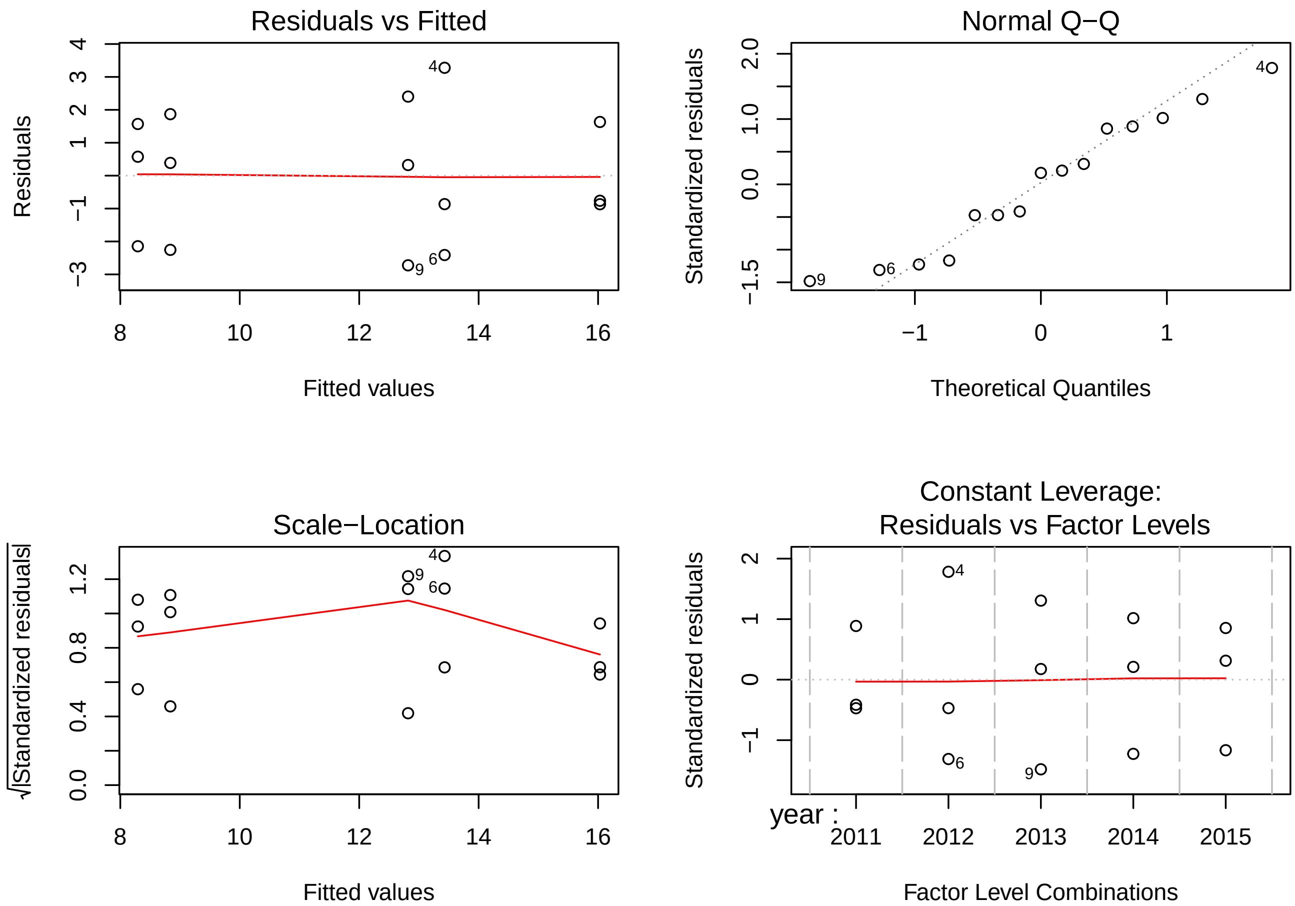
```
#AOV
lm.occ <- aov(ol.c.stock ~ year, data=socf)
op <- par()
par(mfrow=c(2,2))
plot(lm.occ)
par(op)
summary(lm.occ)
```

```
##             Df Sum Sq Mean Sq F value  Pr(>F)
## year         4 127.90   31.97   6.305 0.00845 **
## Residuals   10  50.71    5.07
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#lsmeans test and compact letter display
cld.occ <- cld(emmeans(lm.occ, ~ year),
               alpha = 0.05,
               Letters = letters,
               ordered=TRUE, reversed=TRUE)
cld.occ1 <- cld.occ[order(cld.occ$year),]
#save and show table of results
kable(cld.occ1, "latex", booktabs=TRUE) %>%
  kable_styling(c("striped"))
```

| | year | emmean | SE | df | lower.CL | upper.CL | .group |
|---|---|---|---|---|---|---|---|
| 5 | 2011 | 16.031221 | 1.300186 | 10 | 13.134227 | 18.92822 | a |
| 4 | 2012 | 13.429893 | 1.300186 | 10 | 10.532899 | 16.32689 | ab |
| 3 | 2013 | 12.818432 | 1.300186 | 10 | 9.921438 | 15.71543 | ab |
| 2 | 2014 | 8.837899 | 1.300186 | 10 | 5.940904 | 11.73489 | b |
| 1 | 2015 | 8.293452 | 1.300186 | 10 | 5.396457 | 11.19045 | b |

```
#LM
lm.occ <- lm(ol.c.stock ~ year, data=socf)
op <- par()
par(mfrow=c(2,2))
plot(lm.occ)
```



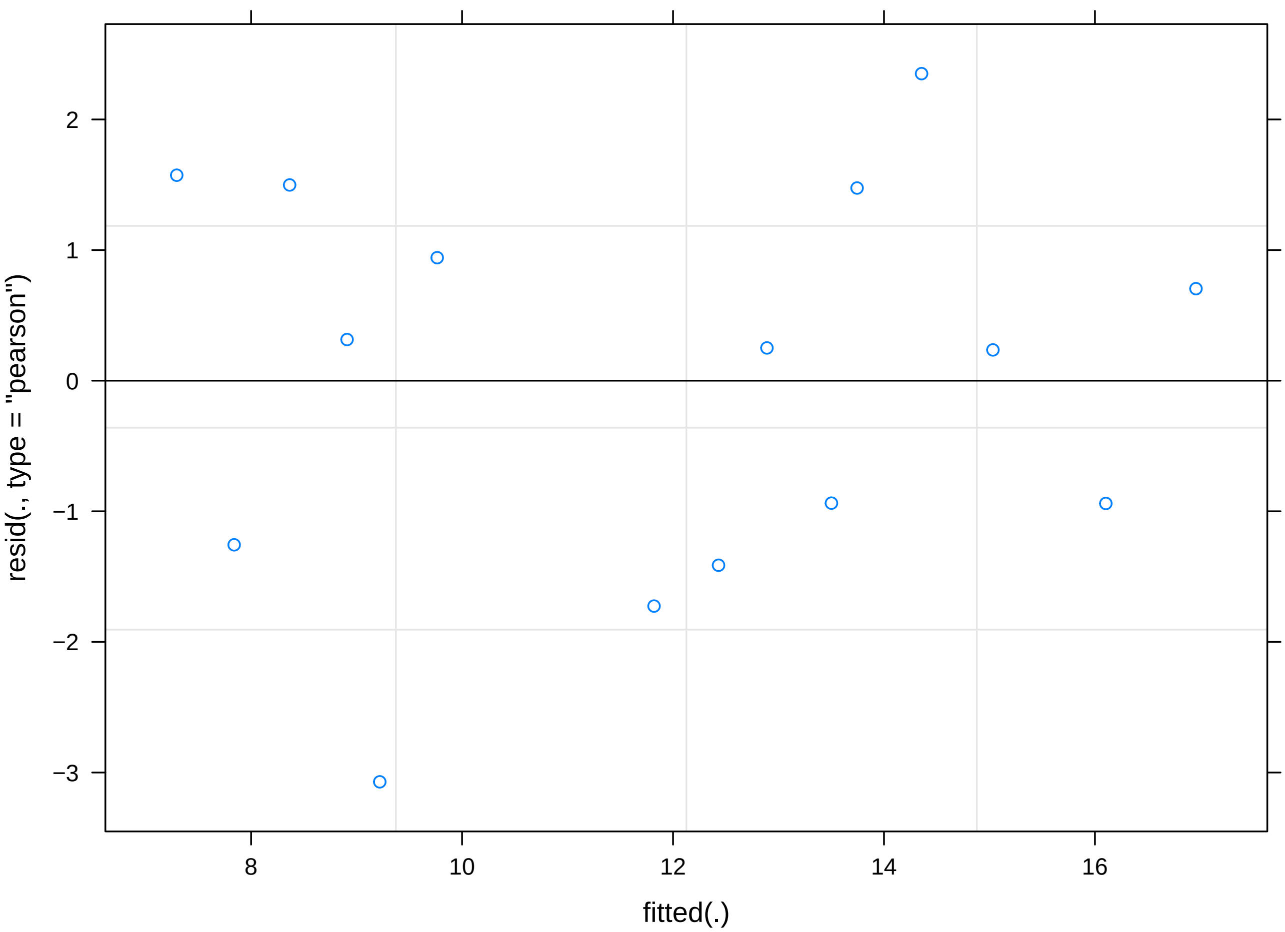

```r
par(op)
summary(lm.occ)
```

```
##
## Call:
## lm(formula = ol.c.stock ~ year, data = socf)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7242 -1.5064  0.3227  1.6006  3.2772
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   16.031      1.300  12.330 2.26e-07 ***
## year2012      -2.601      1.839  -1.415  0.18752
## year2013      -3.213      1.839  -1.747  0.11117
## year2014      -7.193      1.839  -3.912  0.00290 **
## year2015      -7.738      1.839  -4.208  0.00181 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.252 on 10 degrees of freedom
## Multiple R-squared:  0.7161, Adjusted R-squared:  0.6025
## F-statistic: 6.305 on 4 and 10 DF,  p-value: 0.008452
```

```r
#lsmeans test and compact letter display
cld.occ <- cld(emmeans(lm.occ, ~ year),
               alpha = 0.05,
               Letters = letters,
               ordered=TRUE, reversed=TRUE)
cld.occ2 <- cld.occ[order(cld.occ$year),]
#save and show table of results
kable(cld.occ2, "latex", booktabs=TRUE) %>%
  kable_styling(c("striped"))
```

| | year | emmean | SE | df | lower.CL | upper.CL | .group |
|---|---|---|---|---|---|---|---|
| 5 | 2011 | 16.031221 | 1.300186 | 10 | 13.134227 | 18.92822 | a |
| 4 | 2012 | 13.429893 | 1.300186 | 10 | 10.532899 | 16.32689 | ab |
| 3 | 2013 | 12.818432 | 1.300186 | 10 | 9.921438 | 15.71543 | ab |
| 2 | 2014 | 8.837899 | 1.300186 | 10 | 5.940904 | 11.73489 | b |
| 1 | 2015 | 8.293452 | 1.300186 | 10 | 5.396457 | 11.19045 | b |

```r
#LMM
lm.occ <- lmer(ol.c.stock ~ year + (1|rep), data=socf)
op <- par()
par(mfrow=c(2,2))
plot(lm.occ)
```



```r
par(op)
summary(lm.occ)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: ol.c.stock ~ year + (1 | rep)
##    Data: socf
##
## REML criterion at convergence: 49
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.6056 -0.5742  0.1310  0.6317  1.2289
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  rep      (Intercept) 1.412    1.188
##  Residual             3.659    1.913
## Number of obs: 15, groups:  rep, 3
##
## Fixed effects:
##             Estimate Std. Error     df t value Pr(>|t|)
## (Intercept)   16.031      1.300  7.632  12.330  2.6e-06 ***
## year2012      -2.601      1.562  8.000  -1.666  0.13437
## year2013      -3.213      1.562  8.000  -2.057  0.07370 .
## year2014      -7.193      1.562  8.000  -4.606  0.00174 **
## year2015      -7.738      1.562  8.000  -4.954  0.00112 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr) yr2012 yr2013 yr2014
## year2012 -0.601
## year2013 -0.601  0.500
## year2014 -0.601  0.500  0.500
## year2015 -0.601  0.500  0.500  0.500
```

```r
#lsmeans test and compact letter display
cld.occ <- cld(emmeans(lm.occ, ~ year),
               alpha = 0.05,
               Letters = letters,
               ordered=TRUE, reversed=TRUE)
cld.occ3 <- cld.occ[order(cld.occ$year),]
#save and show table of results
kable(cld.occ3, "latex", booktabs=TRUE) %>%
  kable_styling(c("striped"))
```

| | year | emmean | SE | df | lower.CL | upper.CL | .group |
|---|---|---|---|---|---|---|---|
| 5 | 2011 | 16.031221 | 1.300186 | 7.632406 | 13.007665 | 19.05478 | a |
| 4 | 2012 | 13.429893 | 1.300186 | 7.632406 | 10.406337 | 16.45345 | ab |
| 3 | 2013 | 12.818432 | 1.300186 | 7.632406 | 9.794876 | 15.84199 | ab |
| 2 | 2014 | 8.837899 | 1.300186 | 7.632406 | 5.814343 | 11.86145 | b |
| 1 | 2015 | 8.293452 | 1.300186 | 7.632406 | 5.269896 | 11.31701 | b |

### 2.2.5 Dense fraction

The dense fraction was evaluated using a LMM with year as a fixed effect and rep as a random effect. As mentioned previously this is a better model than a repeated measures ANOVA, but also as mentioned previously, accounting for the variance of reps has no effect on the model. To deomnstrate an ANOVA, LM, and LMM are shown below.

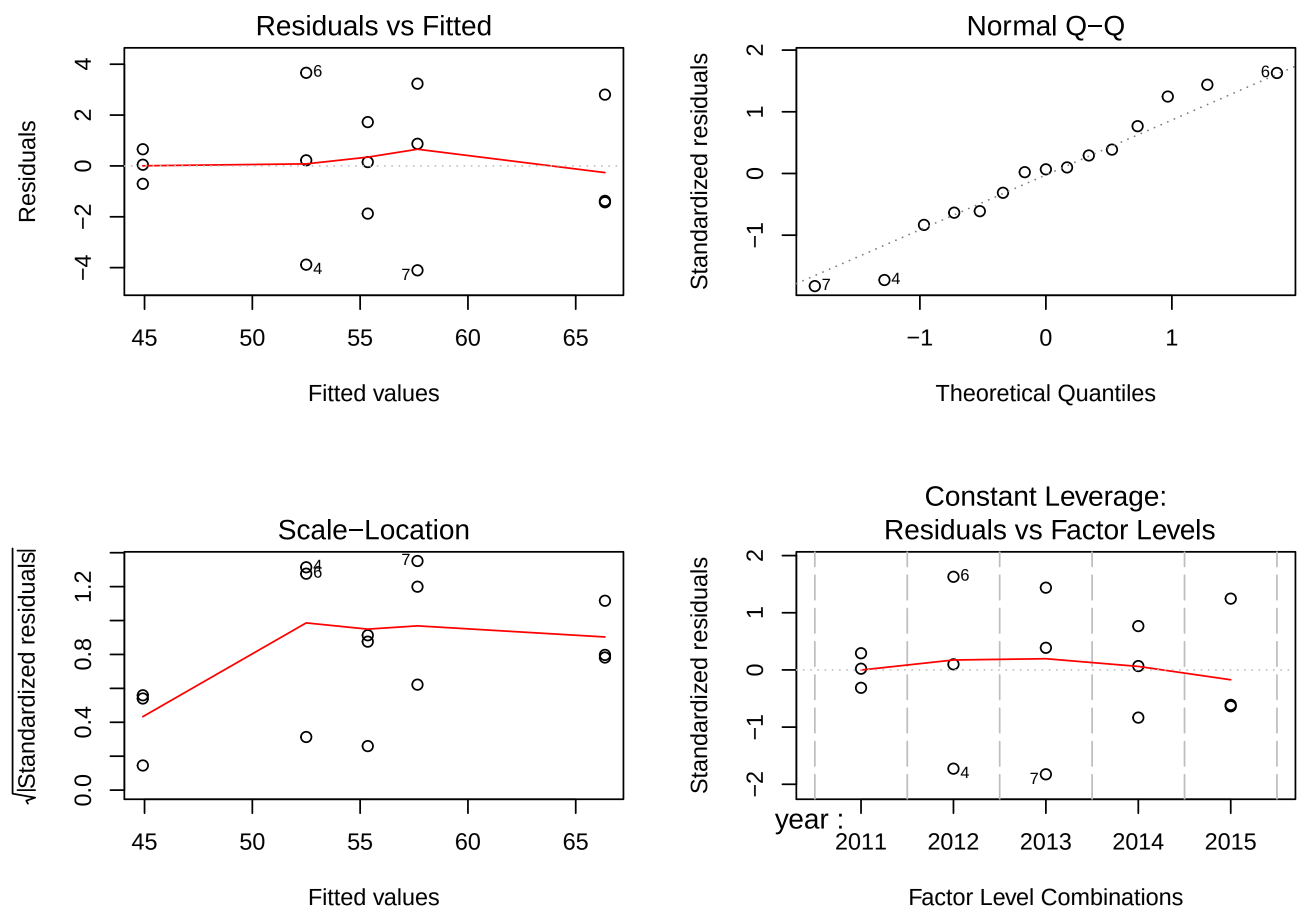
```
#LMM
lm.d <- aov(d.c.stock ~ year, data=socf)
op <- par()
par(mfrow=c(2,2))
plot(lm.d)
par(op)
summary(lm.d)
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)    
## year         4  730.1  182.53   24.07 4.08e-05 ***
## Residuals   10   75.8    7.58                     
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#lsmeans test and compact letter display
cld.d <- cld(emmeans(lm.d, ~ year),
                alpha = 0.05,
                Letters = letters,
                ordered=TRUE, reversed=TRUE)
cld.d1 <- cld.d[order(cld.d$year),]
#save and show table of results
kable(cld.d1, "latex", booktabs=TRUE) %>%
  kable_styling(c("striped"))
```

| | year | emmean | SE | df | lower.CL | upper.CL | .group |
|---|---|---|---|---|---|---|---|
| 5 | 2011 | 44.92008 | 1.590007 | 10 | 41.37732 | 48.46283 | c |
| 3 | 2012 | 52.49860 | 1.590007 | 10 | 48.95584 | 56.04136 | b |
| 2 | 2013 | 57.65968 | 1.590007 | 10 | 54.11692 | 61.20243 | b |
| 4 | 2014 | 55.35428 | 1.590007 | 10 | 51.81152 | 58.89703 | b |
| 1 | 2015 | 66.35555 | 1.590007 | 10 | 62.81279 | 69.89830 | a |

```
#LM
lm.d <- lm(d.c.stock ~ year, data=socf)
op <- par()
par(mfrow=c(2,2))
plot(lm.d)
```



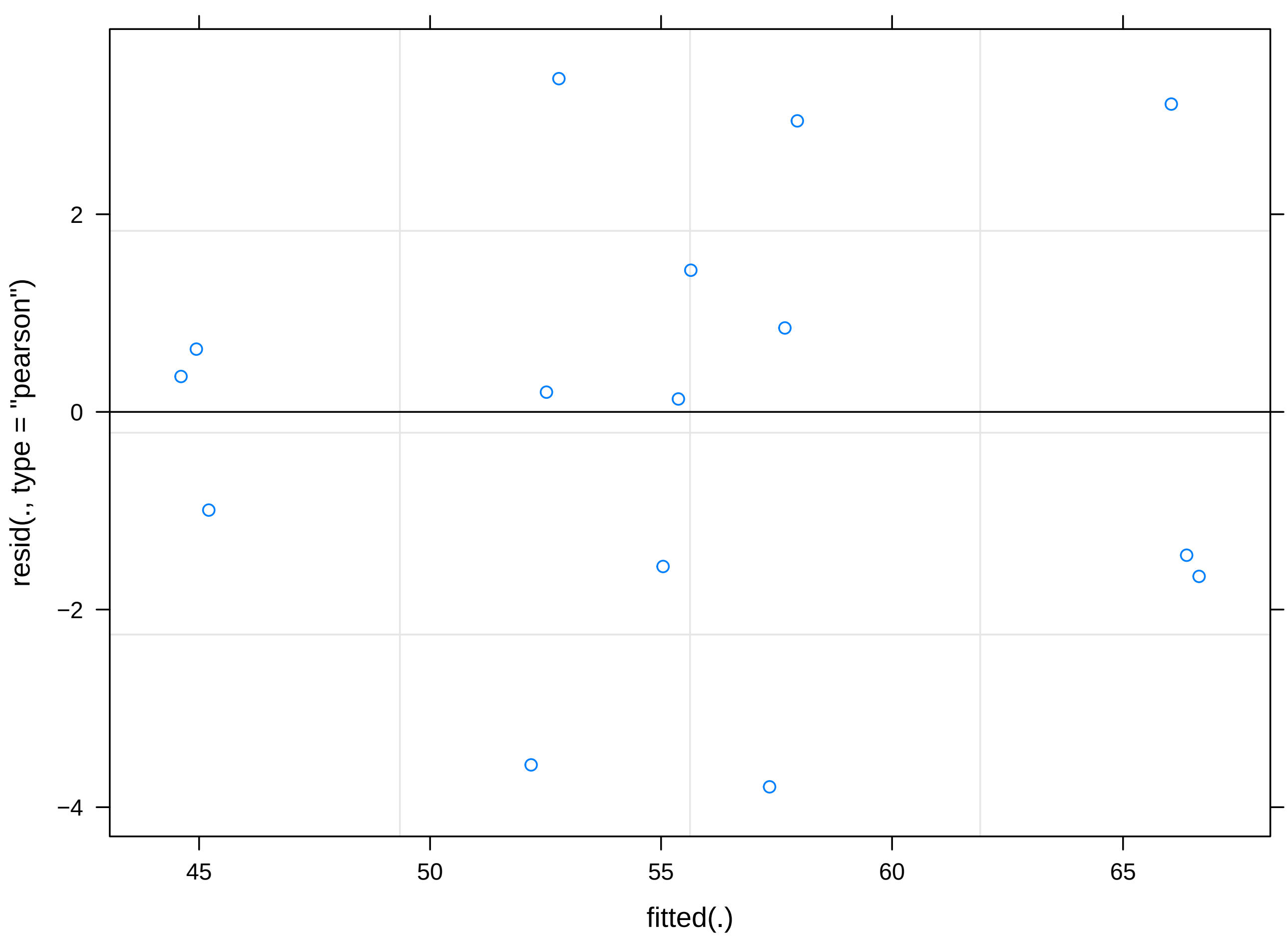

```r
par(op)
summary(lm.d)
```

```
##
## Call:
## lm(formula = d.c.stock ~ year, data = socf)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.1050 -1.4019  0.1514  1.2968  3.6625
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   44.920      1.590  28.252 7.18e-11 ***
## year2012       7.579      2.249   3.370 0.007118 **
## year2013      12.740      2.249   5.666 0.000208 ***
## year2014      10.434      2.249   4.640 0.000922 ***
## year2015      21.435      2.249   9.533 2.46e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.754 on 10 degrees of freedom
## Multiple R-squared:  0.9059, Adjusted R-squared:  0.8683
## F-statistic: 24.07 on 4 and 10 DF,  p-value: 4.081e-05
```

```r
#lsmeans test and compact letter display
cld.d <- cld(emmeans(lm.d, ~ year),
                alpha = 0.05,
                Letters = letters,
                ordered=TRUE, reversed=TRUE)
cld.d2 <- cld.d[order(cld.d$year),]
#save and show table of results
kable(cld.d2, "latex", booktabs=TRUE) %>%
  kable_styling(c("striped"))
```

| | year | emmean | SE | df | lower.CL | upper.CL | .group |
|---|---|---|---|---|---|---|---|
| 5 | 2011 | 44.92008 | 1.590007 | 10 | 41.37732 | 48.46283 | c |
| 3 | 2012 | 52.49860 | 1.590007 | 10 | 48.95584 | 56.04136 | b |
| 2 | 2013 | 57.65968 | 1.590007 | 10 | 54.11692 | 61.20243 | b |
| 4 | 2014 | 55.35428 | 1.590007 | 10 | 51.81152 | 58.89703 | b |
| 1 | 2015 | 66.35555 | 1.590007 | 10 | 62.81279 | 69.89830 | a |

```r
#LMM
lm.d <- lmer(d.c.stock ~ year + (1|rep), data=socf)
op <- par()
par(mfrow=c(2,2))
plot(lm.d)
```



```r
par(op)
summary(lm.d)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: d.c.stock ~ year + (1 | rep)
##    Data: socf
##
## REML criterion at convergence: 54.1
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.41637 -0.56264  0.07456  0.42608  1.25894
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  rep      (Intercept) 0.4088   0.6394
##  Residual             7.1755   2.6787
## Number of obs: 15, groups:  rep, 3
##
## Fixed effects:
##             Estimate Std. Error     df t value Pr(>|t|)
## (Intercept)   44.920      1.590  9.885  28.252 8.78e-11 ***
## year2012       7.579      2.187  8.000   3.465 0.008505 **
## year2013      12.740      2.187  8.000   5.825 0.000394 ***
## year2014      10.434      2.187  8.000   4.771 0.001407 **
## year2015      21.435      2.187  8.000   9.801 9.86e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr) yr2012 yr2013 yr2014
## year2012 -0.688
## year2013 -0.688  0.500
## year2014 -0.688  0.500  0.500
## year2015 -0.688  0.500  0.500  0.500
```

```r
#lsmeans test and compact letter display
cld.d <- cld(emmeans(lm.d, ~ year),
                alpha = 0.05,
                Letters = letters,
                ordered=TRUE, reversed=TRUE)
cld.d3 <- cld.d[order(cld.d$year),]
#save and show table of results
kable(cld.d3, "latex", booktabs=TRUE) %>%
  kable_styling(c("striped"))
```

| | year | emmean | SE | df | lower.CL | upper.CL | .group |
|---|---|---|---|---|---|---|---|
| 5 | 2011 | 44.92008 | 1.590006 | 9.885114 | 41.37174 | 48.46842 | c |
| 3 | 2012 | 52.49860 | 1.590006 | 9.885114 | 48.95026 | 56.04694 | b |
| 2 | 2013 | 57.65968 | 1.590006 | 9.885114 | 54.11133 | 61.20802 | b |
| 4 | 2014 | 55.35428 | 1.590006 | 9.885114 | 51.80593 | 58.90262 | b |
| 1 | 2015 | 66.35555 | 1.590006 | 9.885114 | 62.80721 | 69.90389 | a |

```r
#create a workbook for all results to be store by sheet
wb <- createWorkbook()
addWorksheet(wb, "Root_LM_Depth Contrasts")
addWorksheet(wb, "AGY_LM_Year Contrasts")
addWorksheet(wb, "SOC_GLS_ESM.3600_Yr Contrasts")
addWorksheet(wb, "SOC_GLS_ESM.18000_Yr Contrasts")
addWorksheet(wb, "Fracs_LM_Freel_Year Contrasts")
addWorksheet(wb, "Fracs_LM_Occl_Year Contrasts")
addWorksheet(wb, "Fracs_LM_Dense_Year Contrasts")
#write data to sheets
writeData(wb, sheet="Root_LM_Depth Contrasts", cld.root.d)
writeData(wb, sheet="AGY_LM_Year Contrasts", cld.agy3)
writeData(wb, sheet="SOC_GLS_ESM.3600_Yr Contrasts", cld.soc3)
writeData(wb, sheet="SOC_GLS_ESM.18000_Yr Contrasts", cld.socd3)
writeData(wb, sheet="Fracs_LM_Freel_Year Contrasts", cld.free3)
writeData(wb, sheet="Fracs_LM_Occl_Year Contrasts", cld.occ3)
writeData(wb, sheet="Fracs_LM_Dense_Year Contrasts", cld.d3)
saveWorkbook(wb, file= paste0(statsDir, "/GCB Statistical analyses_revised.xlsx"), overwrite = TRUE)
```

### 2.2.6 Final Modeling results

Here I present all the ANOVA, LM, and LMM results:

```
#list for analysis type
a.t.agy <- data.frame(matrix(c("ANOVA", "ANOVA", "ANOVA", "ANOVA",
            "LM", "LM", "LM", "LM",
            "LMM", "LMM", "LMM", "LMM"), ncol = 1))
colnames(a.t.agy) <- "Analysis Type"
a.t <- data.frame(matrix(c("ANOVA", "ANOVA", "ANOVA", "ANOVA", "ANOVA",
            "LM", "LM", "LM", "LM", "LM",
            "LMM", "LMM", "LMM", "LMM", "LMM"), ncol=1))
colnames(a.t) <- "Analysis Type"
```

#### 2.2.6.1 Above ground biomass yield

AGY is all the same:

```
#AGY contrasts
kable(cbind(a.t.agy,rbind(cld.agy1, cld.agy2, cld.agy3)),
      "latex", booktabs=T, linesep="") %>%
  kable_styling(latex_option="striped", stripe_index = c(5:8))
```

| Analysis Type | year | emmean | SE | df | lower.CL | upper.CL | .group |
|---|---|---|---|---|---|---|---|
| ANOVA | 2012 | 16.66340 | 1.984678 | 8 | 12.08672 | 21.24008 | a |
| ANOVA | 2013 | 18.22843 | 1.984678 | 8 | 13.65176 | 22.80511 | a |
| ANOVA | 2014 | 18.96551 | 1.984678 | 8 | 14.38884 | 23.54219 | a |
| ANOVA | 2015 | 17.32307 | 1.984678 | 8 | 12.74639 | 21.89975 | a |
| LM | 2012 | 16.66340 | 1.984678 | 8 | 12.08672 | 21.24008 | a |
| LM | 2013 | 18.22843 | 1.984678 | 8 | 13.65176 | 22.80511 | a |
| LM | 2014 | 18.96551 | 1.984678 | 8 | 14.38884 | 23.54219 | a |
| LM | 2015 | 17.32307 | 1.984678 | 8 | 12.74639 | 21.89975 | a |
| LMM | 2012 | 16.66340 | 1.984678 | 8 | 12.08672 | 21.24008 | a |
| LMM | 2013 | 18.22843 | 1.984678 | 8 | 13.65176 | 22.80511 | a |
| LMM | 2014 | 18.96551 | 1.984678 | 8 | 14.38884 | 23.54219 | a |
| LMM | 2015 | 17.32307 | 1.984678 | 8 | 12.74639 | 21.89975 | a |

#### 2.2.6.2 surface SOC

Surface SOC is the same:

```
#Surface SOC contrasts
kable(cbind(a.t, rbind(cld.soc1, cld.soc2, cld.soc3)),
      "latex", booktabs=T, linesep="") %>%
  kable_styling(latex_option="striped", stripe_index = c(6:10))
```

| | Analysis Type | year | emmean | SE | df | lower.CL | upper.CL | .group |
|---|---|---|---|---|---|---|---|---|
| 3 | ANOVA | 2011 | 67.27631 | 3.295203 | 17.000000 | 60.32404 | 74.22858 | a |
| 2 | ANOVA | 2012 | 70.16452 | 6.016190 | 17.000000 | 57.47147 | 82.85758 | a |
| 1 | ANOVA | 2013 | 75.33056 | 6.016190 | 17.000000 | 62.63751 | 88.02361 | a |
| 5 | ANOVA | 2014 | 66.80903 | 6.016190 | 17.000000 | 54.11598 | 79.50208 | a |
| 4 | ANOVA | 2015 | 77.35173 | 6.016190 | 17.000000 | 64.65868 | 90.04478 | a |
| 31 | LM | 2011 | 67.27631 | 3.295203 | 17.000000 | 60.32404 | 74.22858 | a |
| 21 | LM | 2012 | 70.16452 | 6.016190 | 17.000000 | 57.47147 | 82.85758 | a |
| 11 | LM | 2013 | 75.33056 | 6.016190 | 17.000000 | 62.63751 | 88.02361 | a |
| 51 | LM | 2014 | 66.80903 | 6.016190 | 17.000000 | 54.11598 | 79.50208 | a |
| 41 | LM | 2015 | 77.35173 | 6.016190 | 17.000000 | 64.65868 | 90.04478 | a |
| 32 | LMM | 2011 | 67.27631 | 4.093934 | 9.415912 | 58.07717 | 76.47545 | a |
| 22 | LMM | 2012 | 68.74860 | 5.063050 | 15.738521 | 58.00090 | 79.49630 | a |
| 12 | LMM | 2013 | 73.91463 | 5.063050 | 15.738521 | 63.16693 | 84.66233 | a |
| 52 | LMM | 2014 | 65.39310 | 5.063050 | 15.738521 | 54.64541 | 76.14080 | a |
| 42 | LMM | 2015 | 75.93580 | 5.063050 | 15.738521 | 65.18810 | 86.68350 | a |

#### 2.2.6.3 Deep SOC

Deep soil has a significant change when rep is accounted for as a random effect:

```
#Deep SOC contrasts
kable(cbind(a.t, rbind(cld.socd1, cld.socd2, cld.socd3)),
      "latex", booktabs=T, linesep="") %>%
  kable_styling(latex_option="striped", stripe_index = c(6:10))
```

| | Analysis Type | year | emmean | SE | df | lower.CL | upper.CL | .group |
|---|---|---|---|---|---|---|---|---|
| 5 | ANOVA | 2011 | 180.1039 | 7.313243 | 17.000000 | 164.6743 | 195.5335 | a |
| 3 | ANOVA | 2012 | 193.7611 | 13.352093 | 17.000000 | 165.5906 | 221.9316 | a |
| 2 | ANOVA | 2013 | 224.4747 | 13.352093 | 17.000000 | 196.3043 | 252.6452 | a |
| 4 | ANOVA | 2014 | 201.5666 | 13.352093 | 17.000000 | 173.3962 | 229.7371 | a |
| 1 | ANOVA | 2015 | 225.7105 | 13.352093 | 17.000000 | 197.5400 | 253.8809 | a |
| 51 | LM | 2011 | 180.1039 | 7.313243 | 17.000000 | 164.6743 | 195.5335 | a |
| 31 | LM | 2012 | 193.7611 | 13.352093 | 17.000000 | 165.5906 | 221.9316 | a |
| 21 | LM | 2013 | 224.4747 | 13.352093 | 17.000000 | 196.3043 | 252.6452 | a |
| 41 | LM | 2014 | 201.5666 | 13.352093 | 17.000000 | 173.3962 | 229.7371 | a |
| 11 | LM | 2015 | 225.7105 | 13.352093 | 17.000000 | 197.5400 | 253.8809 | a |
| 52 | LMM | 2011 | 180.1039 | 8.839692 | 9.877455 | 160.3747 | 199.8332 | b |
| 32 | LMM | 2012 | 196.5063 | 12.438078 | 16.937809 | 170.2569 | 222.7557 | ab |
| 22 | LMM | 2013 | 227.2199 | 12.438078 | 16.937809 | 200.9705 | 253.4693 | a |
| 42 | LMM | 2014 | 204.3118 | 12.438078 | 16.937809 | 178.0624 | 230.5612 | ab |
| 12 | LMM | 2015 | 228.4557 | 12.438078 | 16.937809 | 202.2063 | 254.7051 | a |

#### 2.2.6.4 Free light fraction

Free light fraction is the same:

```
#free light fraction contrasts
kable(cbind(a.t,rbind(cld.free1, cld.free2, cld.free3)),
      "latex", booktabs=T, linesep="") %>%
  kable_styling(latex_option="striped", stripe_index = c(6:10))
```

| | Analysis Type | year | emmean | SE | df | lower.CL | upper.CL | .group |
|---|---|---|---|---|---|---|---|---|
| 4 | ANOVA | 2011 | 6.325013 | 0.6735842 | 10 | 4.824174 | 7.825852 | a |
| 2 | ANOVA | 2012 | 4.236032 | 0.6735842 | 10 | 2.735192 | 5.736871 | ab |
| 5 | ANOVA | 2013 | 4.852449 | 0.6735842 | 10 | 3.351609 | 6.353288 | ab |
| 1 | ANOVA | 2014 | 2.616855 | 0.6735842 | 10 | 1.116016 | 4.117695 | b |
| 3 | ANOVA | 2015 | 2.702727 | 0.6735842 | 10 | 1.201888 | 4.203566 | b |
| 41 | LM | 2011 | 6.325013 | 0.6735842 | 10 | 4.824174 | 7.825852 | a |
| 21 | LM | 2012 | 4.236032 | 0.6735842 | 10 | 2.735192 | 5.736871 | ab |
| 51 | LM | 2013 | 4.852449 | 0.6735842 | 10 | 3.351609 | 6.353288 | ab |
| 11 | LM | 2014 | 2.616855 | 0.6735842 | 10 | 1.116016 | 4.117695 | b |
| 31 | LM | 2015 | 2.702727 | 0.6735842 | 10 | 1.201888 | 4.203566 | b |
| 42 | LMM | 2011 | 6.325013 | 0.6735842 | 10 | 4.824174 | 7.825852 | a |
| 22 | LMM | 2012 | 4.236032 | 0.6735842 | 10 | 2.735192 | 5.736871 | ab |
| 52 | LMM | 2013 | 4.852449 | 0.6735842 | 10 | 3.351609 | 6.353288 | ab |
| 12 | LMM | 2014 | 2.616855 | 0.6735842 | 10 | 1.116016 | 4.117695 | b |
| 32 | LMM | 2015 | 2.702727 | 0.6735842 | 10 | 1.201888 | 4.203566 | b |

#### 2.2.6.5 Occluded light fraction

Occluded light fraction is the same:

```r
#Occluded light fraction contrasts
kable(cbind(a.t,rbind(cld.occ1, cld.occ2, cld.occ3)),
      "latex", booktabs=T, linesep="") %>%
  kable_styling(latex_option="striped", stripe_index = c(6:10))
```

| | Analysis Type | year | emmean | SE | df | lower.CL | upper.CL | .group |
|---|---|---|---|---|---|---|---|---|
| 5 | ANOVA | 2011 | 16.031221 | 1.300186 | 10.000000 | 13.134227 | 18.92822 | a |
| 4 | ANOVA | 2012 | 13.429893 | 1.300186 | 10.000000 | 10.532899 | 16.32689 | ab |
| 3 | ANOVA | 2013 | 12.818432 | 1.300186 | 10.000000 | 9.921438 | 15.71543 | ab |
| 2 | ANOVA | 2014 | 8.837899 | 1.300186 | 10.000000 | 5.940904 | 11.73489 | b |
| 1 | ANOVA | 2015 | 8.293452 | 1.300186 | 10.000000 | 5.396457 | 11.19045 | b |
| 51 | LM | 2011 | 16.031221 | 1.300186 | 10.000000 | 13.134227 | 18.92822 | a |
| 41 | LM | 2012 | 13.429893 | 1.300186 | 10.000000 | 10.532899 | 16.32689 | ab |
| 31 | LM | 2013 | 12.818432 | 1.300186 | 10.000000 | 9.921438 | 15.71543 | ab |
| 21 | LM | 2014 | 8.837899 | 1.300186 | 10.000000 | 5.940904 | 11.73489 | b |
| 11 | LM | 2015 | 8.293452 | 1.300186 | 10.000000 | 5.396457 | 11.19045 | b |
| 52 | LMM | 2011 | 16.031221 | 1.300186 | 7.632406 | 13.007665 | 19.05478 | a |
| 42 | LMM | 2012 | 13.429893 | 1.300186 | 7.632406 | 10.406337 | 16.45345 | ab |
| 32 | LMM | 2013 | 12.818432 | 1.300186 | 7.632406 | 9.794876 | 15.84199 | ab |
| 22 | LMM | 2014 | 8.837899 | 1.300186 | 7.632406 | 5.814343 | 11.86145 | b |
| 12 | LMM | 2015 | 8.293452 | 1.300186 | 7.632406 | 5.269896 | 11.31701 | b |

#### 2.2.6.6 Dense fraction

Dense fraction is the same:

```r
#Dense fraction contrasts
kable(cbind(a.t,rbind(cld.d1, cld.d2, cld.d3)),
      "latex", booktabs=T, linesep="") %>%
  kable_styling(latex_option="striped", stripe_index = c(6:10))
```

| | Analysis Type | year | emmean | SE | df | lower.CL | upper.CL | .group |
|---|---|---|---|---|---|---|---|---|
| 5 | ANOVA | 2011 | 44.92008 | 1.590007 | 10.000000 | 41.37732 | 48.46283 | c |
| 3 | ANOVA | 2012 | 52.49860 | 1.590007 | 10.000000 | 48.95584 | 56.04136 | b |
| 2 | ANOVA | 2013 | 57.65968 | 1.590007 | 10.000000 | 54.11692 | 61.20243 | b |
| 4 | ANOVA | 2014 | 55.35428 | 1.590007 | 10.000000 | 51.81152 | 58.89703 | b |
| 1 | ANOVA | 2015 | 66.35555 | 1.590007 | 10.000000 | 62.81279 | 69.89830 | a |
| 51 | LM | 2011 | 44.92008 | 1.590007 | 10.000000 | 41.37732 | 48.46283 | c |
| 31 | LM | 2012 | 52.49860 | 1.590007 | 10.000000 | 48.95584 | 56.04136 | b |
| 21 | LM | 2013 | 57.65968 | 1.590007 | 10.000000 | 54.11692 | 61.20243 | b |
| 41 | LM | 2014 | 55.35428 | 1.590007 | 10.000000 | 51.81152 | 58.89703 | b |
| 11 | LM | 2015 | 66.35555 | 1.590007 | 10.000000 | 62.81279 | 69.89830 | a |
| 52 | LMM | 2011 | 44.92008 | 1.590006 | 9.885114 | 41.37174 | 48.46842 | c |
| 32 | LMM | 2012 | 52.49860 | 1.590006 | 9.885114 | 48.95026 | 56.04694 | b |
| 22 | LMM | 2013 | 57.65968 | 1.590006 | 9.885114 | 54.11133 | 61.20802 | b |
| 42 | LMM | 2014 | 55.35428 | 1.590006 | 9.885114 | 51.80593 | 58.90262 | b |
| 12 | LMM | 2015 | 66.35555 | 1.590006 | 9.885114 | 62.80721 | 69.90389 | a |

# 3 Compartment model

Setup data for model:

```
#set baseline and year vector to model
baseline <- fractions[1,c(5,7,9)]
years <- seq(0,10,by=0.1)

#subset each carbon fraction and carbon totals
fracs <-    fractions
free <-     data.frame(time=fracs$year, Ct.1=fracs$freeLFmean, sdC1=fracs$freeLFsd)
occluded <- data.frame(time=fracs$year, Ct.2=fracs$occLFmean,  sdC2=fracs$occLFsd)
dense <-    data.frame(time=fracs$year, Ct.3=fracs$DFmean,     sdC3=fracs$DFsd)
total <-    data.frame(time=fracs$year, Ct.4=fracs$CstockMgha, sdC4=fracs$Cstocksd)

#Inputs (Mg-C/ha) from 20cm soil core data (2020.02.11 GCB combined Data)
max.in <- 1.05
mean.in <- 0.83
min.in <- 0.62

#Cinput ~1.8 is too low to fit; by ~3.5 models starts to over-estimate
Cinput <- 2.50
```

Initial model parameters are set and fraction model defined:

- Initial K1 set to mean value of root input (0.83 Mg-C/ha)
  - Initial k2 set to half input
  - Initial k3 set to 1/10 input
- All alpha values set to 50% as an initial guess

We know that carbon flux out of the soil and root inputs are related and similar in scale which is why carbon stocks in a system tend toward an equilibrium and are slow to change in undisturbed systems. Here we make the initial model guess that k1 is about the same size as the measured root input, while k2 and k3 decay rates are scaled down by 1/5 and 1/10 respectively. This is a simple starting point for the model, based on observed data, as it searches for an optimized solution.

```
#Initial parameters
inipars<-c(k1=mean.in,
           k2=mean.in/5,
           k3=mean.in/10,
           alpha21=0.5,
           alpha12=0.5,
           alpha32=0.5,
           alpha23=0.5)

ctrlModel <- function(pars,In,C0) {
  mod=ThreepFeedbackModel(t=years, ks=c(pars[1:3]), C0=C0, In=In,
                          a21=(pars[4])*pars[1],
                          a12=(pars[5])*(pars[2]-pars[6]*pars[2]),
                          a32=(pars[6])*pars[2],
                          a23=(pars[7])*pars[3],
                          pass=TRUE)
  Ct=getC(mod)
  return(data.frame(time=years, Ct=Ct))
}

#Cost function
fracs_cost <- function(pars){
  modOut=ctrlModel(pars, In=Cinput, C0=as.numeric(baseline))
```

```
  cost1=modCost( model=modOut,obs=free,    err="sdC1",x="time")
  cost2=modCost( model=modOut,obs=occluded,err="sdC2",x="time",cost=cost1)
  return(modCost(model=modOut,obs=dense,   err="sdC3",x="time",cost=cost2))
}
```

## 3.1 Classical model evaluation

Model is classically fit using Nelder-Mead method. Input parameters are controlled within estimates developed from root data:

- Model values for K1 are controlled relative to root input estimates
  - Max estimate (mean + SE) of yearly root input was 1.05 Mg-C/ha
  - Min estimate (mean - SE) of yearly root input was 0.62 Mg-C/ha
- It is unlikely that k1 is greater than 4 times the maximum measured root input (4 x 1.05 = 4.2 Mg-C/ha) or there would likely be declining C
- Similarly, it is unlikely that free root decay is less than 1/2 min inputs (0.5 x 0.62 = 0.31 Mg-C/ha) or larger increases in C would be observed
- K2 and K3 were controlled between 0 - Inf as we don't have data to better control these values
- Alpha values controlled between 0 - 1

The above model controls do influence the final model fits but are in place to keep input and k1 decay rates reasonable. The fraction data and the compartment model structure have a greater influence on final values than these broad restrictions on the parameter solution space. Here we simply constrain the model parameters within a range that makes sense compared to observed root data. The solution space for the model to test/estimate best parameters is still large, just controlled somewhat so the model can't converge on unreasonable estimates.

The Nelder-mead method of model fitting implemented in FME::modFit estimates a singular solution below ~2.0 Mg-C/ha with initial conditions and k1 limitations described above. This is the first indication that the system needs at least 2 Mg-C/ha inputs to see the C increases observed in the soil fractionation data; anything lower than 2.0 Mg-C/ha in inputs and the model struggles to find a solution.

```
#Use FME modfit function
frac_fit<-modFit(f=fracs_cost,
                  p=inipars,
                  method="Nelder-Mead",
                  lower=c(min.in/2,0,0,0,0,0,0),
                  upper=c(max.in*4,Inf,Inf,1,1,1,1))

#Set all par[5] values back to alpha1,2
class.par <- fracs_fit$par
class.par[5] <- (class.par[5]*(class.par[2]-class.par[6]*class.par[2]))/class.par[2]
fracs_fit$par <- class.par

#look at model parameters
class.par %>%
  kable("latex", booktabs=TRUE) %>%
    kable_styling(c("striped"))
```

| | x |
|---|---|
| k1 | 0.9715023 |
| k2 | 0.4176363 |
| k3 | 0.0002653 |
| alpha21 | 0.8641529 |
| alpha12 | 0.0205687 |
| alpha32 | 0.8710036 |
| alpha23 | 0.1170978 |

```
#Load best classically optimized parameters into model to calculate carbon
bestmod<-ThreepFeedbackModel(
  t=years, In=Cinput,
  C0=as.numeric(baseline),
  ks=c(class.par[1:3]),
```

```
  a21=class.par[4]*class.par[1],
  a12=class.par[5]*class.par[2],
  a32=class.par[6]*class.par[2],
  a23=class.par[7]*class.par[3], pass=TRUE)

#calculate carbon
classical.Ct <- getC(bestmod)
classical.ctotal <- rowSums(classical.Ct)
classical.Ct <- cbind(classical.Ct,classical.ctotal)
```

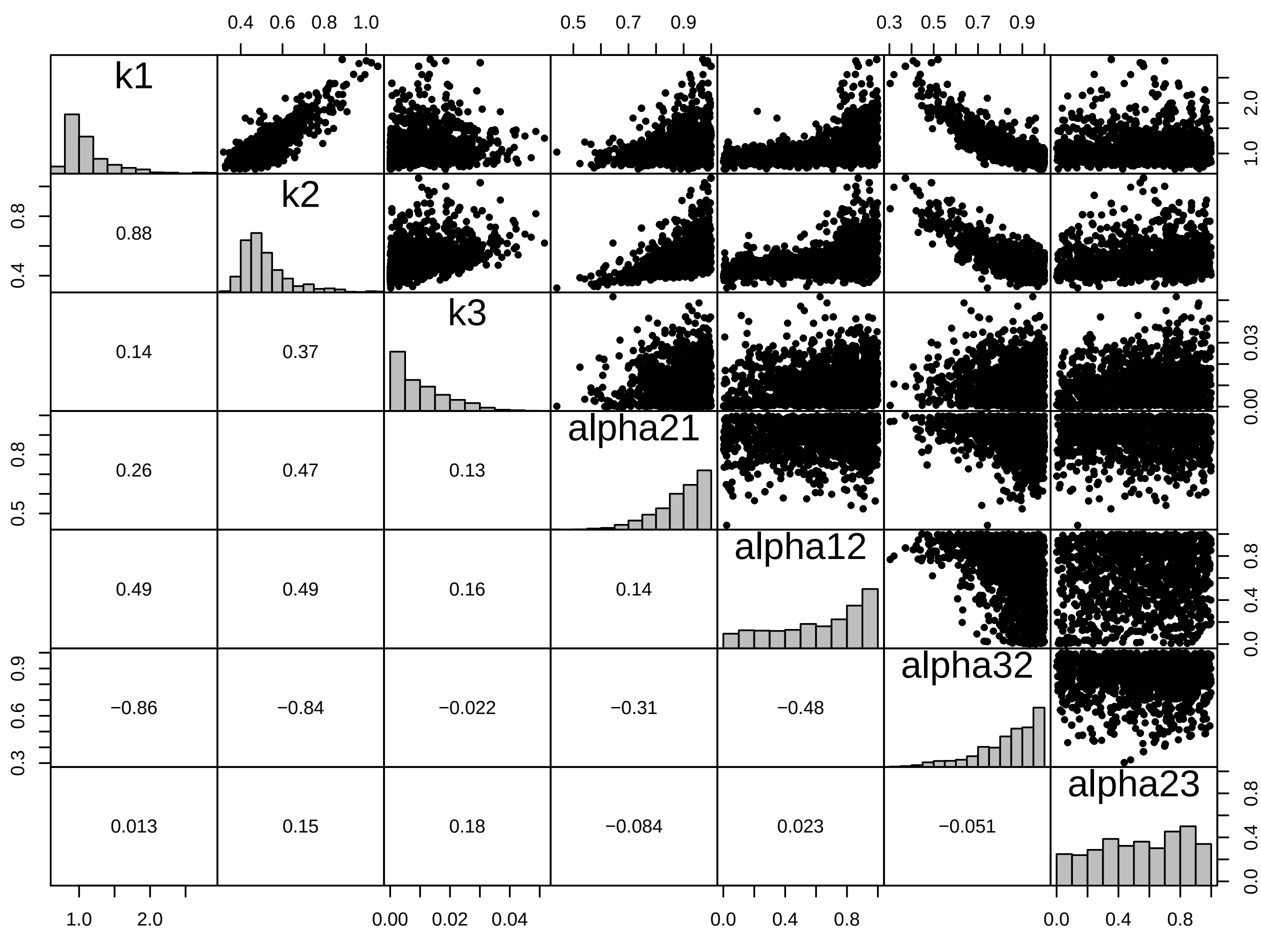
## 3.2 Bayesian model evaluation

Markov Chain Monte Carlo evaluation of model to estimate model error:

```
#Using MCMC to test model
var0 <- fracs_fit$var_ms
fracs_mcmc <- modMCMC(f=fracs_cost, p=fracs_fit$par, jump=fracs_fit$par*0.05,
                      var0=var0, wvar0=1, updatecov=50,
                      niter=40e3,
                      lower=c(min.in/2,0,0,0,0,0,0),
                      upper=c(max.in*4,Inf,Inf,1,1,1,1))
```

```
## number of accepted runs: 3760 out of 40000 (9.4%)
```

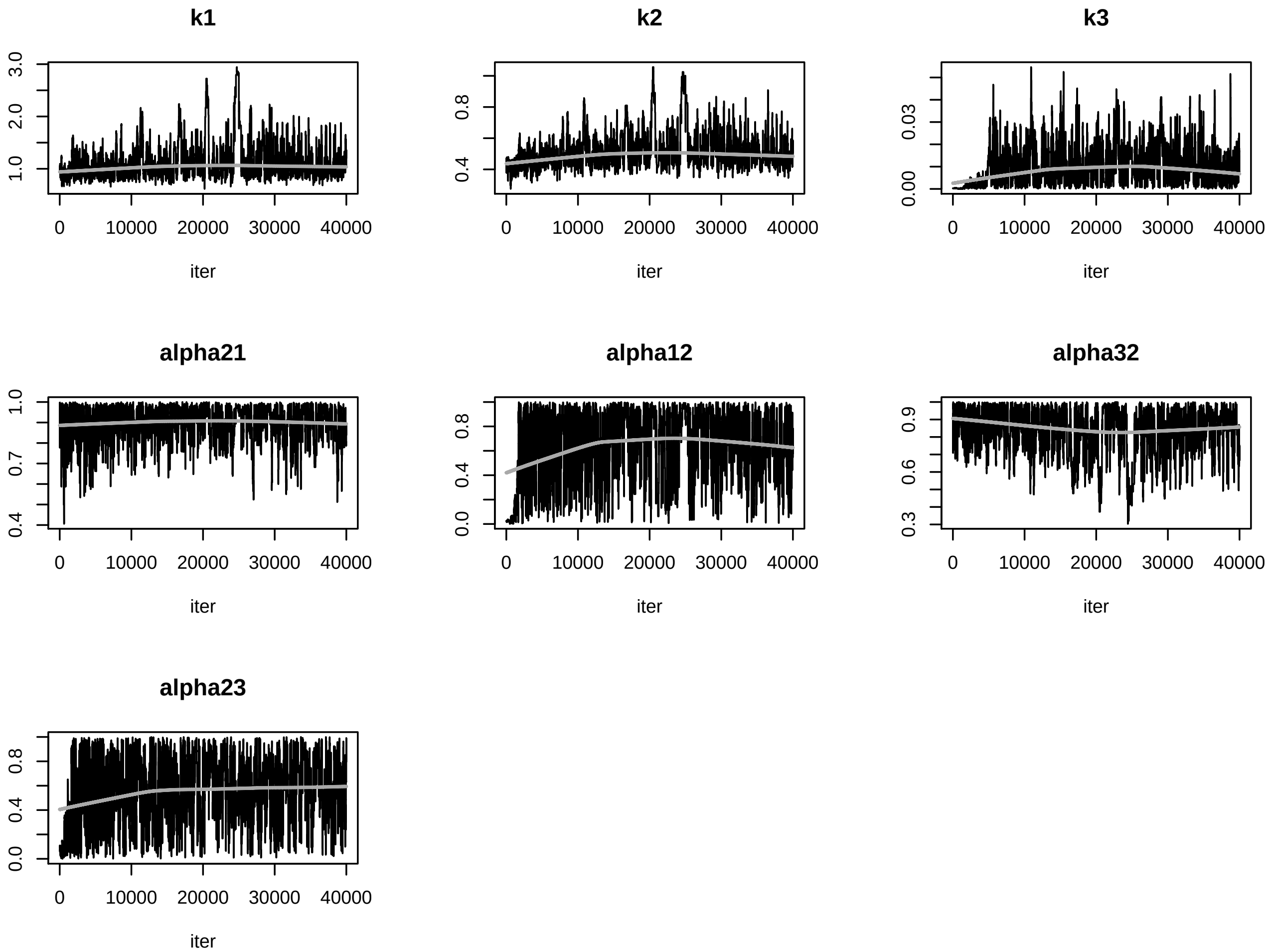
```
#susbet data to reduce points in pairs plot
frac_mcmc_sub <- fracs_mcmc
fracs_mcmc_sub[["pars"]] <- fracs_mcmc_sub[["pars"]][sample(nrow(fracs_mcmc$pars), 2000, replace=FALSE),]
pairs(fracs_mcmc_sub, pch=20)
```




```
plot(fracs_mcmc)

#Bayesian best fit parameters
mcmc.par <- fracs_mcmc
m.par <- fracs_mcmc$pars
#Set all par[5] values back to alpha1,2
mcmc.par$pars[,5] <- (m.par[,5]* (m.par[,2]-m.par[,6]*m.par[,2]))/m.par[,2]
#combine classical and bayesian parameter estimates
```

```r
fracs_sum <- rbind(class.par,summary(mcmc.par)[,c(1:7)])

#Set best parameters from MCMC fit as model parameters
mcmc <- colMeans(mcmc.par$pars)
mcmc_mod <- ThreepFeedbackModel(t=years, In=Cinput,
                                C0=as.numeric(baseline),
                                ks=c(mcmc[1:3]),
                                a21=mcmc[4]*mcmc[1],
                                a12=mcmc[5]*mcmc[2],
                                a32=mcmc[6]*mcmc[2],
                                a23=mcmc[7]*mcmc[3], pass=TRUE)

#Calculate carbon changes from Bayesian fit
mcmc.Ct <- getC(mcmc_mod)
mcmc.ctotal <- rowSums(mcmc.Ct)

#look at model
summary(mcmc.par)[1:7] %>%
  kable("latex", booktabs=TRUE) %>%
    kable_styling("striped")
```

| | k1 | k2 | k3 | alpha21 | alpha12 | alpha32 | alpha23 |
|---|---|---|---|---|---|---|---|
| mean | 1.1390688 | 0.5207836 | 0.0100529 | 0.8900127 | 0.1263534 | 0.8317877 | 0.5470756 |
| sd | 0.3753442 | 0.1216180 | 0.0091324 | 0.0898245 | 0.1340372 | 0.1409853 | 0.2781808 |
| min | 0.6150611 | 0.2747122 | 0.0000014 | 0.4057290 | 0.0000083 | 0.3033448 | 0.0000603 |
| max | 2.9441808 | 1.0560904 | 0.0546658 | 0.9998507 | 0.5541204 | 0.9999149 | 0.9998401 |
| q025 | 0.8985233 | 0.4401132 | 0.0026834 | 0.8412847 | 0.0206316 | 0.7506035 | 0.3245156 |
| q050 | 1.0166256 | 0.4879597 | 0.0076326 | 0.9120535 | 0.0738274 | 0.8688353 | 0.5660782 |
| q075 | 1.2472650 | 0.5628786 | 0.0147433 | 0.9603892 | 0.1926759 | 0.9451876 | 0.7902124 |


k1
k2
k3
alpha21
alpha12
alpha32
alpha23
iter
0
10000
20000
30000
40000
1.0
2.0
3.0
0.4
0.8
0.00
0.03
0.7
0.0
0.3
0.6
0.9

### 3.3 Graphing final figure

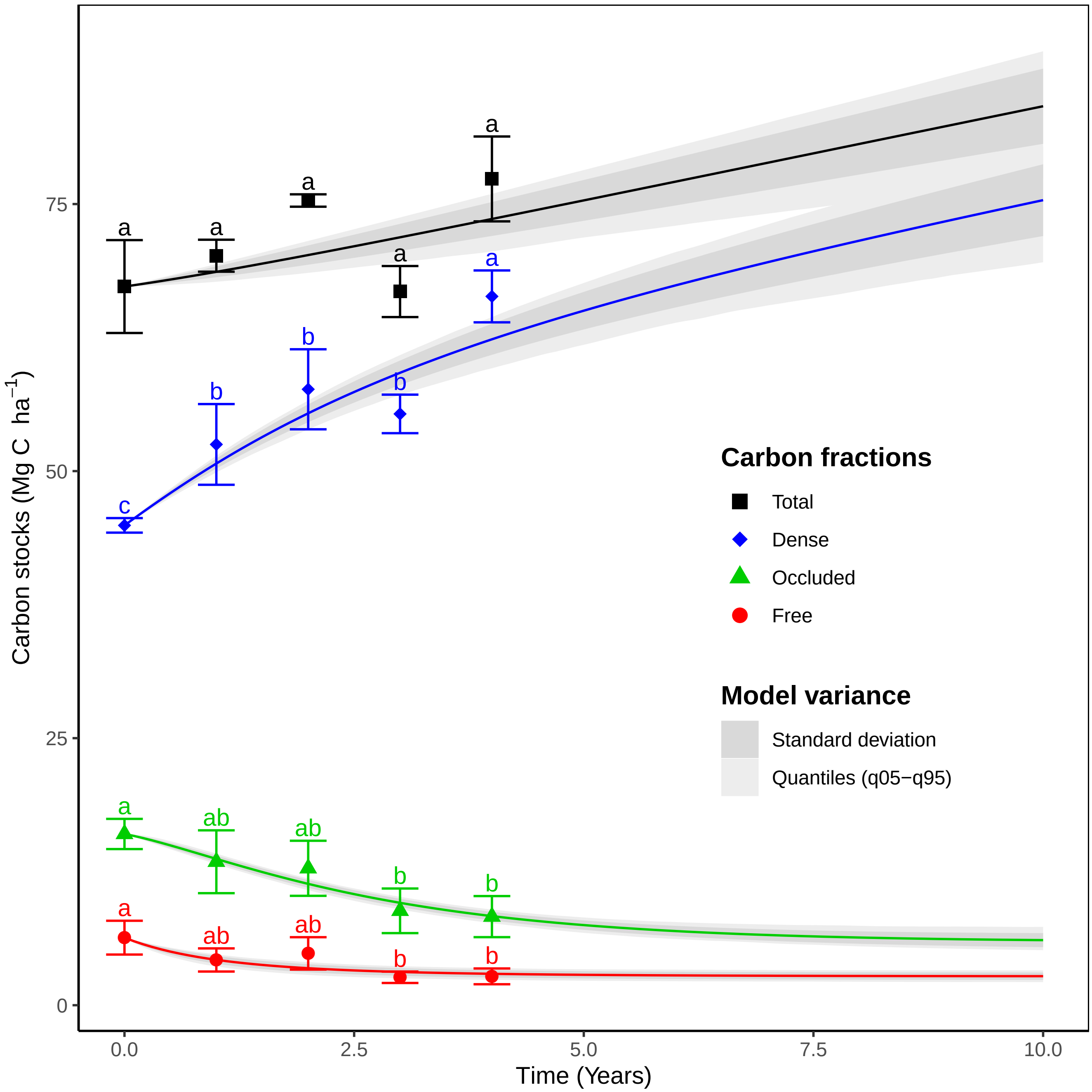
The final figure for publication is drawn below from fitted MCMC model:

```
#plot for publication - ggplot
#add Cinput to parameter vector
pars <- cbind(mcmc.par$pars, Cinput)

#Function to optimize - remade with input as part of parameter
ctrlModel3 <- function(pars) {
  mod=ThreepFeedbackModel(t=years, ks=c(pars[1:3]), C0=as.numeric(baseline),
                          In=pars[8],
                          a21=(pars[4])*pars[1],
                          a12=(pars[5])*pars[2],
                          a32=(pars[6])*pars[2],
                          a23=(pars[7])*pars[3],
                          pass=TRUE)
  Ct=getC(mod)
  Ct=cbind(Ct,rowSums(Ct))
  return(data.frame(time=years, Ct=Ct))
}

#Use sensRange(FME package) to test variability of MCMC parameter fitting
ct.1 <- summary(sensRange(num=1000, func=ctrlModel3, parInput=pars,
                          sensvar=c("Ct.1")))
ct.2 <- summary(sensRange(num=1000, func=ctrlModel3, parInput=pars,
                          sensvar=c("Ct.2")))
ct.3 <- summary(sensRange(num=1000, func=ctrlModel3, parInput=pars,
                          sensvar=c("Ct.3")))
ct.4 <- summary(sensRange(num=1000, func=ctrlModel3, parInput=pars,
                          sensvar=c("Ct.4")))

#add column of fraction names to use during merge with point data
ct.1$name <- "Ct.1"
ct.2$name <- "Ct.2"
ct.3$name <- "Ct.3"
ct.4$name <- "Ct.4"

#stack all sensRange results from each fraction into one long dataframe
cts <- rbind(ct.1, ct.2, ct.3, ct.4)

#melt to organize dataframe
lines <- melt(cts, id=c("name", "x", "Mean", "Sd", "Min", "Max",
                        "q05", "q25", "q50", "q75", "q95"))

#cleaning sensRange output
names(lines)[2] <- "time"

#Create dataframe of measured data to merge into graphing dataframe
points <- cbind(free[,c(1:2)],occluded[,c(1:2)],dense[,c(1:2)],total[,c(1:2)])
points_sd <- cbind(free[,c(1,3)],occluded[,c(1,3)],dense[,c(1,3)],total[,c(1,3)])
pt <- melt(points, id=c("time"))
pt$sd <- melt(points_sd, id=c("time"))[,3]
pt <- pt[,c(2,1,3:4)]
names(pt) <- c("name","time","pt.mean", "pt.sd")
pt[,1] <- as.character(pt[,1])

#merge modeled lines and measured point data
```

```r
all.data <- merge(lines, pt, by=intersect(names(lines),names(pt)),
                  all=TRUE, sort=TRUE)
all.data$name <- as.factor(all.data$name)
levels(all.data$name) <- c("Free", "Occluded", "Dense", "Total")

#complete cases to obtain only pt data
pt.data <- all.data[complete.cases(all.data),-c(3:11)]
#add significance letters to dataframe for plotting
pt.data$sig <- trimws(c(cld.free3$.group, cld.occ3$.group, cld.d3$.group, cld.soc3$.group))
#calculate y value to place letters, add 1.25 buffer space
pt.data$pos <- pt.data$pt.mean + pt.data$pt.sd + 1.25
#create vector to color the added letters
pt.data$col <- c(rep(2,5), rep(3,5), rep(4,5), rep(1,5))

#plot model results/variance and measured data
pdf(paste0(mainDir,"/Figures/Final publication figures/fractions.pdf"))
ggplot(all.data, aes(x=time, y=Mean, col=name)) +
  geom_ribbon(aes(ymin=q05, ymax=q95, group=name, fill="Quantiles (q05-q95)"), colour=NA) +
  geom_ribbon(aes(ymin=Mean-Sd, ymax=Mean+Sd, group=name, fill="Standard deviation"), colour=NA) +
  scale_fill_manual(values=c("grey93","grey85")) +
  geom_line(show.legend = FALSE) + scale_color_manual(values=c(2,3,4,1)) +
  coord_cartesian(xlim = c(0, 10)) + theme_bw() + theme(panel.border = element_rect(color="black"),
                                                        panel.grid.major = element_blank(), panel.grid.mino
                                                        legend.title = element_text(colour="black", size=12
  geom_point(aes(x=time, y=pt.mean, col=name, shape=name), size=2.5) + scale_shape_manual(values=c(16,17,18
  geom_errorbar(aes(ymin=pt.mean-pt.sd, ymax=pt.mean+pt.sd), width=0.4) +
  guides(shape=guide_legend(reverse=TRUE, title="Carbon fractions",order=1,override.aes=list(colour=c(1,4,3
         color=guide_legend(reverse=TRUE, title="Carbon fractions",order=1, override.aes=list(linetype=0)),
         fill=guide_legend(reverse=TRUE, title="Model variance", order=2, override.aes=list(colour=c("grey8
                                                                                            fill=c("grey85"
  xlab("Time (Years)") + ylab(expression(paste("Carbon stocks (Mg C  ", ha^-1, ")"))) +
  annotate("text", x=pt.data$time, y=pt.data$pos, label=pt.data$sig, color=pt.data$col)
dev.off()
```

```
## pdf
##   2
```

```r
knitr::include_graphics(paste0(mainDir,"/Figures/Final publication figures/fractions.pdf"))
```



Carbon stocks (Mg C ha$^{-1}$)
Time (Years)
0
25
50
75
0.0
2.5
5.0
7.5
10.0
Carbon fractions
Total
Dense
Occluded
Free
Model variance
Standard deviation
Quantiles (q05−q95)

## 3.4 Final model parameters

Final model parameters:

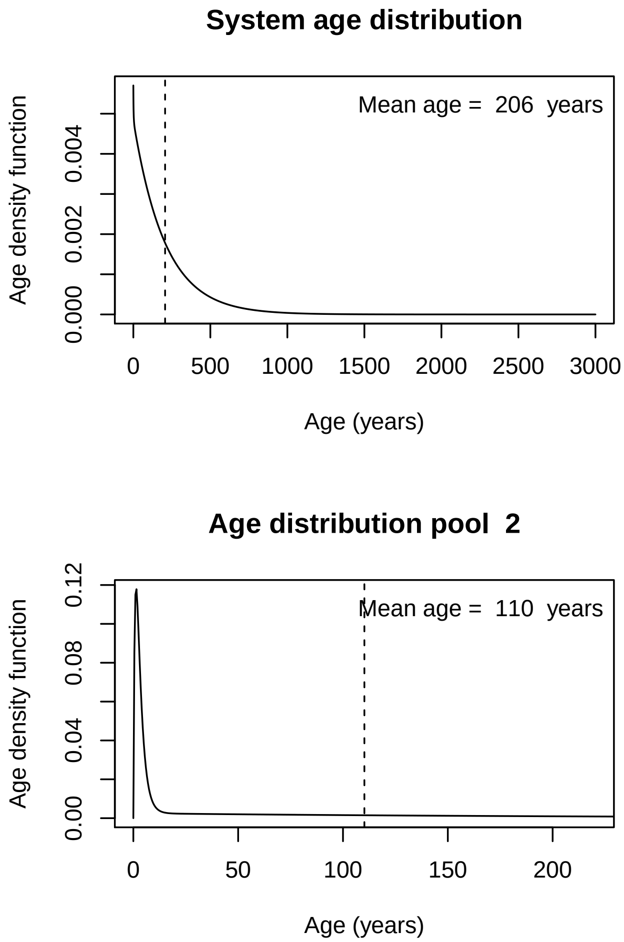
```
#combine classical and bayesian parameter estimates
fracs_sum <- rbind(class.par,summary(mcmc.par)[,c(1:7)])
#get best paramaters from classical and bayesian parameter fits
c.b.par <- fracs_sum[1,]
b.b.par <- fracs_sum[2,]

#rename columns to classical and bayesian
rownames(fracs_sum)[c(1:2)] <- c("Classical","Bayesian")

#create xtable and output to excel
modelpars=xtable(fracs_sum[c(1:8),])
write.xlsx(modelpars, paste0(mainDir,dataDir,"/R_output_modelpars.xlsx"),
                    row.names=TRUE)

#table for knitr output
fracs_sum[c(1:8),] %>%
  kable("latex",booktabs=TRUE) %>%
    kable_styling("striped")
```

| | k1 | k2 | k3 | alpha21 | alpha12 | alpha32 | alpha23 |
|---|---|---|---|---|---|---|---|
| Classical | 0.9715023 | 0.4176363 | 0.0002653 | 0.8641529 | 0.0205687 | 0.8710036 | 0.1170978 |
| Bayesian | 1.1390688 | 0.5207836 | 0.0100529 | 0.8900127 | 0.1263534 | 0.8317877 | 0.5470756 |
| sd | 0.3753442 | 0.1216180 | 0.0091324 | 0.0898245 | 0.1340372 | 0.1409853 | 0.2781808 |
| min | 0.6150611 | 0.2747122 | 0.0000014 | 0.4057290 | 0.0000083 | 0.3033448 | 0.0000603 |
| max | 2.9441808 | 1.0560904 | 0.0546658 | 0.9998507 | 0.5541204 | 0.9999149 | 0.9998401 |
| q025 | 0.8985233 | 0.4401132 | 0.0026834 | 0.8412847 | 0.0206316 | 0.7506035 | 0.3245156 |
| q050 | 1.0166256 | 0.4879597 | 0.0076326 | 0.9120535 | 0.0738274 | 0.8688353 | 0.5660782 |
| q075 | 1.2472650 | 0.5628786 | 0.0147433 | 0.9603892 | 0.1926759 | 0.9451876 | 0.7902124 |

### 3.5 Residence and transfer rates

Carbon residence times and transfer rates are calculated:

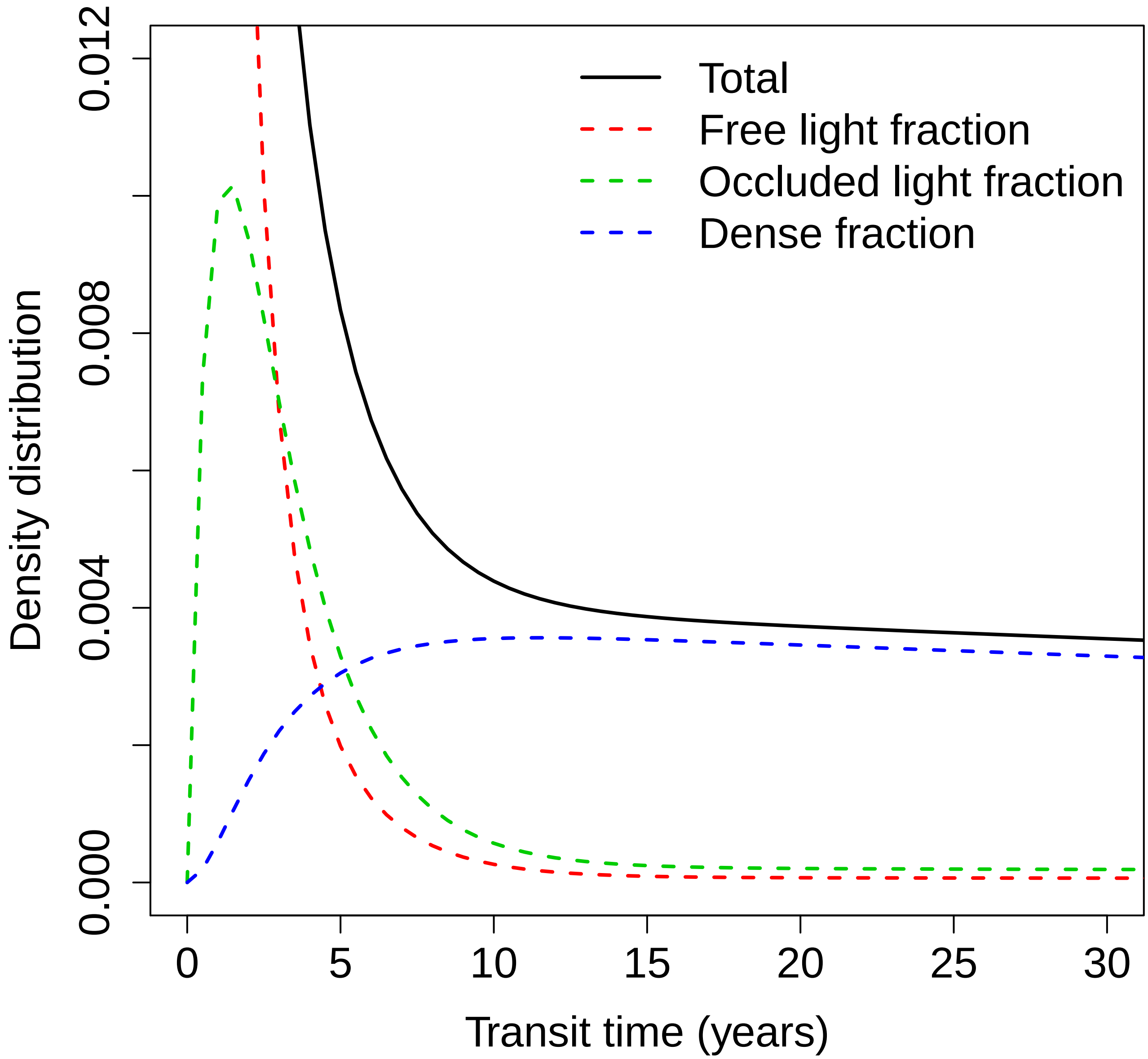
```r
#input matrix
In <- matrix(c(Cinput,0,0), ncol=1)

#make bayesian A matrix
b.A <- diag(-b.b.par[,1:3])
b.A[2,1]=b.b.par[1,4]*b.b.par[1,1]
b.A[1,2]=b.b.par[1,5]*b.b.par[1,2]
b.A[3,2]=b.b.par[1,6]*b.b.par[1,2]
b.A[2,3]=b.b.par[1,7]*b.b.par[1,3]
bz=-1*colSums(b.A)
bx=-solve(b.A)%*%In
br=bz*bx

# sequence of number for the computation of the distribution
tau=seq(0,3000, by=0.5)
#calculate the system age and transit time for classical and bayesian solutions
b.ages=systemAge(A=b.A, u=In, a=tau, q=c(0.5))
b.transitTimes=transitTime(A=b.A, u=In, a=tau, q=c(0.5))

par(mfrow=c(2,2))
plot(tau, b.ages$systemAgeDensity, type="l",
     xlab="Age (years)", ylab="Age density function",
     main="System age distribution")
abline(v=b.ages$meanSystemAge, lty=2)
legend("topright", legend=paste("Mean age = ",
                                round(b.ages$meanSystemAge), " years"), bty="n")
xlims=c(b.ages$meanPoolAge[1,1],b.ages$meanPoolAge[2,1],b.ages$meanPoolAge[3,1])
for(i in 1:3){
  plot(tau, b.ages$poolAgeDensity[,i], type="l",
       xlab="Age (years)", ylab="Age density function",
       main=paste("Age distribution pool ", i), xlim=c(0, xlims[i]*2))
  abline(v=b.ages$meanPoolAge[i,1], lty=2)
  legend("topright", legend=paste("Mean age = ",
                                  round(b.ages$meanPoolAge[i,1]), " years"), bty="n")
}
```




```r
par(mfrow=c(1,1))

# Bayesian Transit time distribution
pdf(paste0(mainDir,"/Figures/Final publication figures/TransitTime.pdf"))
plot(tau, b.transitTimes$transitTimeDensity, type="l", lwd=2, cex.lab=1.5, cex.axis=1.5,
     xlim=c(0,30), ylim=c(0,0.012),
     xlab="Transit time (years)",
     ylab="Density distribution")
#abline(v=b.transitTimes$meanTransitTime, lty=2)
matlines(tau,(b.ages$poolAgeDensity%*%diag(as.numeric(br/sum(br)))), col=2:4, lwd=2, lty=2)
legend("topright", c("Total", "Free light fraction", "Occluded light fraction", "Dense fraction"), col=1:4,
dev.off()
```

```
## pdf
##   2
```

```r
knitr::include_graphics(paste0(mainDir, "/Figures/Final publication figures/TransitTime.pdf"))
```




```r
#create a small table to compare classical/bayesian transit times
transit <- data.frame(b.transitTimes$meanTransitTime)
colnames(transit)[1] <- "Transit Time (Years)"
rownames(transit)[c(1)] <- c("Bayesian")

#table using kable
round(transit,0) %>%
  kable("latex",booktabs=TRUE) %>%
    kable_styling(c("striped"))
```

| | Transit Time (Years) |
|---|---|
| Bayesian | 175 |

# 4 Running the model with varied inputs

Here I repeat the above fitting using the same estimates of root inputs to control K1 (the free SOM decay constant) while changing the C input for each new model within the range that works (below ~1.8 won't fit; above ~3.5 the estimate of total C starts to visibly over-estimate). Graphs are not created as the final graph is usually very similar. Larger changes occur in the underlying decay and transfer rates as the model fits adjust for differences in inputs so that the final C pools still fit the observed changes in free light, Occluded light, and dense SOC fractions. I collect the MCMC model parameters for each input tested, calculate the transfer times of each pool, and combine all into a dataframe to investigate parameter response to input differences.

I ran the model from 1.80 to 5.00 by 0.05 Mg-C/ha increments:

```
#function to calculate MCMC model parameters and transit rates
par.transit.fun <- function(fracs, input){

#set baseline and year vector to model
baseline <- fractions[1,c(5,7,9)]
years <- seq(0,10,by=0.1)

#subset each carbon fraction and carbon totals
free <-     data.frame(time=fracs$year, Ct.1=fracs$freeLFmean, sdC1=fracs$freeLFsd)
occluded <- data.frame(time=fracs$year, Ct.2=fracs$occLFmean,  sdC2=fracs$occLFsd)
dense <-    data.frame(time=fracs$year, Ct.3=fracs$DFmean,     sdC3=fracs$DFsd)
total <-    data.frame(time=fracs$year, Ct.4=fracs$CstockMgha, sdC4=fracs$Cstocksd)

#Inputs (Mg-C/ha) from 20cm soil core data (2020.02.11 GCB combined Data)
max.in <- 1.05
mean.in <- 0.83
min.in <- 0.62

#Cinput ~1.75 is too low to fit; by ~3.2 models starts to over-estimate
Cinput <- input

#Initial parameters
inipars<-c(k1=mean.in,
           k2=mean.in/5,
           k3=mean.in/10,
           alpha21=0.5,
           alpha12=0.5,
           alpha32=0.5,
           alpha23=0.5)

ctrlModel <- function(pars,In,C0) {
  mod=ThreepFeedbackModel(t=years, ks=c(pars[1:3]), C0=C0, In=In,
                          a21=(pars[4])*pars[1],
                          a12=(pars[5])*(pars[2]-pars[6]*pars[2]),
                          a32=(pars[6])*pars[2],
                          a23=(pars[7])*pars[3],
                          pass=TRUE)
  Ct=getC(mod)
  return(data.frame(time=years, Ct=Ct))
}

#Cost function
fracs_cost <- function(pars){
  modOut=ctrlModel(pars, In=Cinput, C0=as.numeric(baseline))
  cost1=modCost( model=modOut,obs=free,    err="sdC1",x="time")
  cost2=modCost( model=modOut,obs=occluded,err="sdC2",x="time",cost=cost1)
```

```r
  return(modCost(model=modOut,obs=dense,   err="sdC3",x="time",cost=cost2))
}

#Use FME modfit function
fracs_fit<-modFit(f=fracs_cost,
                  p=inipars,
                  method="Nelder-Mead",
                  lower=c(min.in/2,0,0,0,0,0,0),
                  upper=c(max.in*4,Inf,Inf,1,1,1,1))

#Using MCMC to test model
var0 <- fracs_fit$var_ms
fracs_mcmc <- modMCMC(f=fracs_cost, p=fracs_fit$par, jump=fracs_fit$par*0.05,
                      var0=var0, wvar0=1, updatecov=50,
                      niter=40e3,
                      lower=c(min.in/2,0,0,0,0,0,0),
                      upper=c(max.in*4,Inf,Inf,1,1,1,1))

#Bayesian best fit parameters
mcmc.par <- fracs_mcmc
m.par <- fracs_mcmc$pars
#Set all par[5] values back to alpha1,2
mcmc.par$pars[,5] <- (m.par[,5]* (m.par[,2]-m.par[,6]*m.par[,2]))/m.par[,2]

#Set best parameters from MCMC fit as model parameters
mcmc <- colMeans(mcmc.par$pars)
mcmc_mod <- ThreepFeedbackModel(t=years, In=Cinput,
                                C0=as.numeric(baseline),
                                ks=c(mcmc[1:3]),
                                a21=mcmc[4]*mcmc[1],
                                a12=mcmc[5]*mcmc[2],
                                a32=mcmc[6]*mcmc[2],
                                a23=mcmc[7]*mcmc[3], pass=TRUE)
#calculate final pool values
Ct <- data.frame(getC(mcmc_mod))
Ct <- Ct[c(101),]
Ct$X4 <- Ct[,1] + Ct[,2] + Ct[,3]
colnames(Ct)[1:4] <- c("10 yr FreeL C", "10 yr OccL C",
                       "10 yr Dense C", "10 yr Total C")

#get best part from MCMC
b.b.par <- summary(mcmc.par)[1,c(1:7)]
#input matrix
In <- matrix(c(Cinput,0,0), ncol=1)
#make bayesian A matrix
b.A <- diag(-b.b.par[,1:3])
b.A[2,1]=b.b.par[1,4]*b.b.par[1,1]
b.A[1,2]=b.b.par[1,5]*b.b.par[1,2]
b.A[3,2]=b.b.par[1,6]*b.b.par[1,2]
b.A[2,3]=b.b.par[1,7]*b.b.par[1,3]
bz=-1*colSums(b.A)
bx=-solve(b.A)%*%In
br=bz*bx
# sequence of number for the computation of the distribution
tau=seq(0,3000, by=0.5)
#calculate the system age and transit time for classical and bayesian solutions
```

```r
b.ages=systemAge(A=b.A, u=In, a=tau)
b.transitTimes=transitTime(A=b.A, u=In, a=tau)

b.a <- data.frame(t(b.ages$meanPoolAge))
colnames(b.a)[1:3] <- c("P1 avg age", "P2 avg age", "P3 avg age")
b.t <- data.frame(b.transitTimes$meanTransitTime)
colnames(b.t)[1] <- c("Mean transit time")
model.outputs <- data.frame(round(Cinput,2), b.b.par, b.a, b.t, Ct)
return(list("MCMC.fit"=mcmc.par, "Final.pars"=model.outputs))
}

#create a dataframe for the results
#namevector <- c("Cinput", "k1", "k2", "k3",
#                 "alpha21", "alpha12", "alpha32", "alpha23",
#                 "P1 avg age", "P2 avg age", "P3 avg age",
#                 "Mean transit time" )
#input.tests <- setNames(data.frame(matrix(ncol = 12, nrow = 0)), namevector)
#input.tests[,1:12] <- sapply(input.tests[,1:12], as.numeric)

 input.tests <- list()
#loop through inputs starting at 1.8
i <- 1.80
#track dataframe column
j <- 1
while(i <= 5.00){
  #save inputs
  input.tests[[j]] <- par.transit.fun(fractions,round(i,2))
  #update input by 0.1; update row by 1
  i <- i + 0.05
  j <- j + 1
}

#save mcmc and model outputs
saveRDS(input.tests, paste0(mainDir, dataDir, "/R_mcmc_output.rds"))

#pull second object from every list to obtain final parameters
library(data.table)
mod.pars <- lapply(input.tests, `[[`, 2)
mod.pars <- data.frame(rbindlist(mod.pars))

#output table of parameters from input tests
modelpars2=xtable(mod.pars)
write.xlsx(modelpars2, paste0(mainDir,dataDir,"/R_output_modelpars_input_tests.xlsx"),
                    row.names=FALSE)

#summary table for knitr output
data.frame(do.call(cbind, lapply(mod.pars, summary))) %>%
  round(digits=3) %>%
  kable("latex", booktabs=TRUE) %>%
    kable_styling(c("striped"), full_width = TRUE)
```

### 4.1 2.5 Mg C/ha/yr model

I used this section to look at the graphs for a range of C inputs to the model. If we want to re-graph anything we can pull the RDS file of mcmc output and excel file of C input and best parameters to create a figure from any model tested (1.8-5.0 Mg C/ha/yr by 0.05). For the final markdown document I'm just re-graphing the 2.5 Mg C/ha/yr model for both the final figure and pool densities:

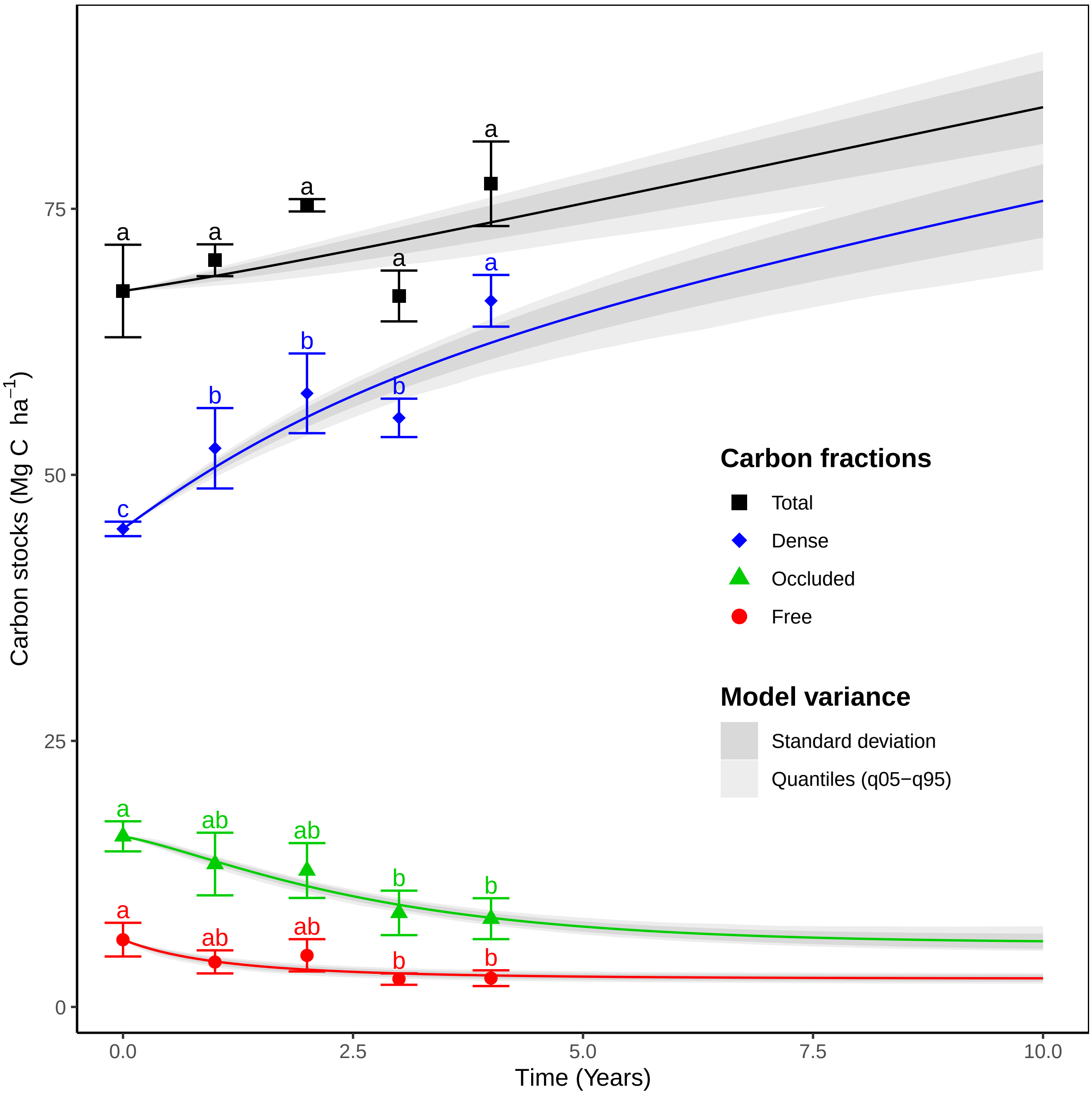
```
#function to graph from model input tests
graph.pars <-  function(mcmc.pars, m.pars, pt, fractions, pt.data) {

#set mcmc output to pars for sensrange
pars <- cbind(mcmc.pars[["pars"]], "Cinput" = as.numeric(m.pars[1,8]))

#set baseline and year vector to model
baseline <- fractions[1,c(5,7,9)]
years <- seq(0,10,by=0.1)

#Function to optimize - remade with input as part of parameter
ctrlModel3 <- function(pars) {
  mod=ThreepFeedbackModel(t=years, ks=c(pars[1:3]), C0=as.numeric(baseline),
                          In=pars[8],
                          a21=(pars[4])*pars[1],
                          a12=(pars[5])*pars[2],
                          a32=(pars[6])*pars[2],
                          a23=(pars[7])*pars[3],
                          pass=TRUE)
  Ct=getC(mod)
  Ct=cbind(Ct,rowSums(Ct))
  return(data.frame(time=years, Ct=Ct))
}

#Use sensRange(FME package) to test variability of MCMC parameter fitting
ct.1 <- summary(sensRange(num=1000, func=ctrlModel3, parInput=pars,
                          sensvar=c("Ct.1")))
ct.2 <- summary(sensRange(num=1000, func=ctrlModel3, parInput=pars,
                          sensvar=c("Ct.2")))
ct.3 <- summary(sensRange(num=1000, func=ctrlModel3, parInput=pars,
                          sensvar=c("Ct.3")))
ct.4 <- summary(sensRange(num=1000, func=ctrlModel3, parInput=pars,
                          sensvar=c("Ct.4")))

#add column of fraction names to use during merge with point data
ct.1$name <- "Ct.1"
ct.2$name <- "Ct.2"
ct.3$name <- "Ct.3"
ct.4$name <- "Ct.4"

#stack all sensRange results from each fraction into one long dataframe
cts <- rbind(ct.1, ct.2, ct.3, ct.4)

#melt to organize dataframe
lines <- melt(cts, id=c("name", "x", "Mean", "Sd", "Min", "Max",
                        "q05", "q25", "q50", "q75", "q95"))

#cleaning sensRange output
names(lines)[2] <- "time"

#merge modeled lines and measured point data
```

```r
all.data <- merge(lines, pt, by=intersect(names(lines),names(pt)),
                  all=TRUE, sort=TRUE)
all.data$name <- as.factor(all.data$name)
levels(all.data$name) <- c("Free", "Occluded", "Dense", "Total")

#plot model results/variance and measured data
g.plot <- ggplot(all.data, aes(x=time, y=Mean, col=name)) +
  geom_ribbon(aes(ymin=q05, ymax=q95, group=name, fill="Quantiles (q05-q95)"), colour=NA) +
  geom_ribbon(aes(ymin=Mean-Sd, ymax=Mean+Sd, group=name, fill="Standard deviation"), colour=NA) +
  scale_fill_manual(values=c("grey93","grey85")) +
  geom_line(show.legend = FALSE) + scale_color_manual(values=c(2,3,4,1)) +
  coord_cartesian(xlim = c(0, 10)) + theme_bw() +
  theme(panel.border = element_rect(color="black"),
  panel.grid.major = element_blank(),
  panel.grid.minor = element_blank(),
  axis.line = element_line(colour = "black"),
  legend.title = element_text(colour="black", size=12, face="bold"),
  legend.position=c(0.75,0.4)) +
  geom_point(aes(x=time, y=pt.mean, col=name, shape=name), size=2.5) +
  scale_shape_manual(values=c(16,17,18,15)) +
  geom_errorbar(aes(ymin=pt.mean-pt.sd, ymax=pt.mean+pt.sd), width=0.4) +
  guides(shape=guide_legend(reverse=TRUE, title="Carbon fractions",
                            order=1,override.aes=list(colour=c(1,4,3,2), size=3)),
         color=guide_legend(reverse=TRUE, title="Carbon fractions",
                            order=1, override.aes=list(linetype=0)),
         fill=guide_legend(reverse=TRUE, title="Model variance",
                           order=2, override.aes=list(colour=c("grey85","grey93"),
        fill=c("grey85","grey93")))) +
  xlab("Time (Years)") +
  ylab(expression(paste("Carbon stocks (Mg C  ", ha^-1, ")"))) +
  annotate("text", x=pt.data$time, y=pt.data$pos, label=pt.data$sig, color=pt.data$col)

pdf(paste0(mainDir,"/Figures/Final publication figures/fractions_", round(m.pars[,8],2),"MgC_input.pdf"))
print(g.plot)
dev.off()

# create transit time graph
#get best part from MCMC
b.b.par <- summary(mcmc.pars)[1,c(1:7)]
#input matrix
In <- matrix(c(as.numeric(m.pars[1,8]),0,0), ncol=1)
#make bayesian A matrix
b.A <- diag(-b.b.par[,1:3])
b.A[2,1]=b.b.par[1,4]*b.b.par[1,1]
b.A[1,2]=b.b.par[1,5]*b.b.par[1,2]
b.A[3,2]=b.b.par[1,6]*b.b.par[1,2]
b.A[2,3]=b.b.par[1,7]*b.b.par[1,3]
bz=-1*colSums(b.A)
bx=-solve(b.A)%*%In
br=bz*bx
# sequence of number for the computation of the distribution
tau=seq(0,3000, by=0.5)
#calculate the system age and transit time for classical and bayesian solutions
b.ages=systemAge(A=b.A, u=In, a=tau)
b.transitTimes=transitTime(A=b.A, u=In, a=tau)
b.age.text <- c(paste0("(", round(b.ages$meanPoolAge[1], 1), " years)"),
```

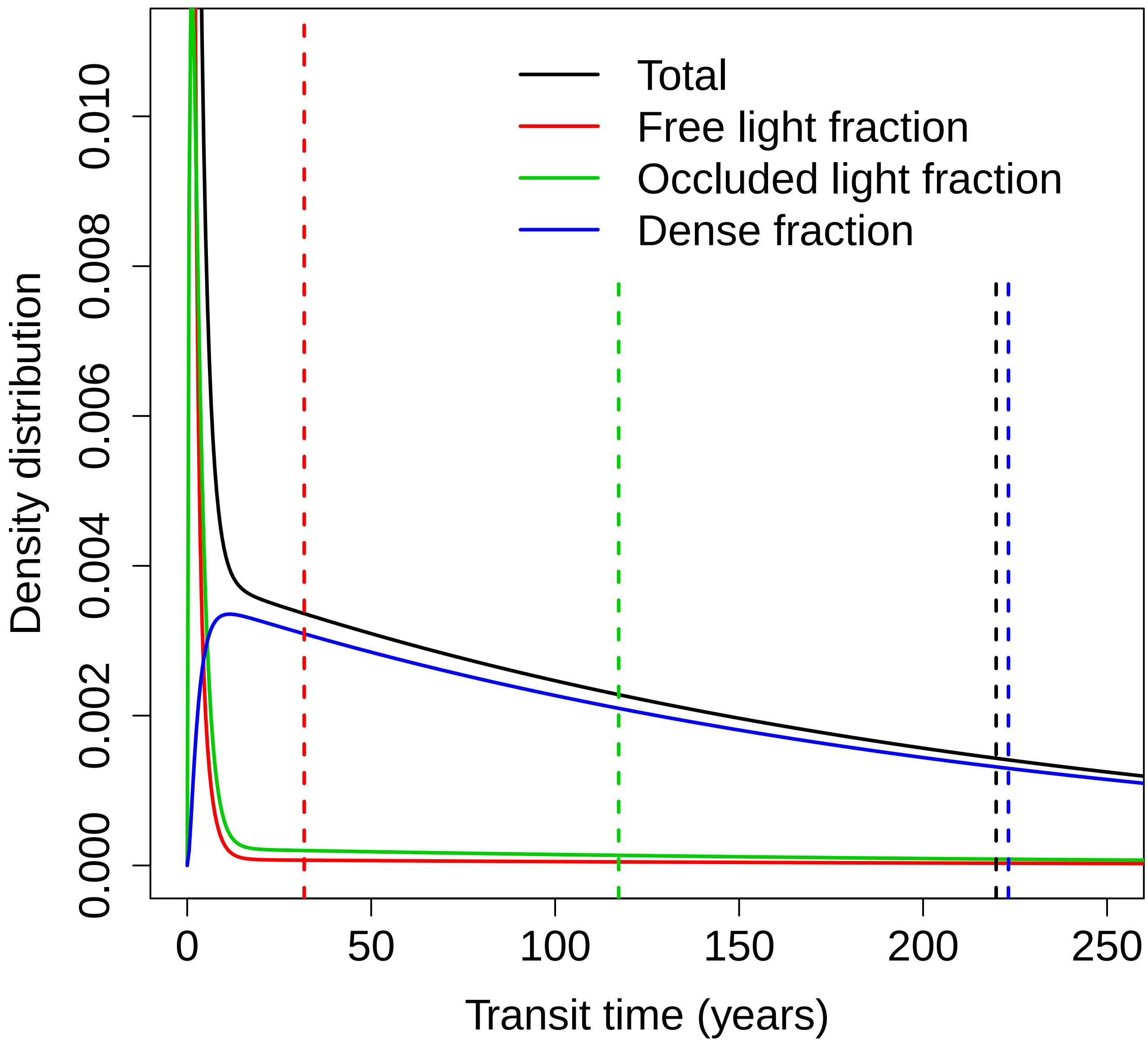
```
                 paste0("(", round(b.ages$meanPoolAge[2], 1), " years)"),
                 paste0("(", round(b.ages$meanPoolAge[3], 1), " years)"),
                 paste0("(", round(b.ages$meanSystemAge, 1), " years)"))
age.x <- c(b.ages$meanPoolAge, b.ages$meanSystemAge)
age.x.t <- c(b.ages$meanPoolAge+c(32,34,37), b.ages$meanSystemAge-32)
age.y <- c(rep(0.0045, 3), 0.0035)
#plot transit times
pdf(paste0(mainDir,"/Figures/Final publication figures/TransitTime_", round(m.pars[,8],2),"MgC_input.pdf"))
par(xpd=FALSE)
plot(tau, b.transitTimes$transitTimeDensity, type="l", lwd=2, lty=1, cex.lab=1.5, cex.axis=1.5,
     xlim=c(0,250), ylim=c(0,0.011),
     xlab="Transit time (years)",
     ylab="Density distribution")
#abline(v=b.transitTimes$meanTransitTime, lty=2)
matlines(tau,(b.ages$poolAgeDensity%*%diag(as.numeric(br/sum(br)))), col=2:4, lwd=2, lty=c(1,1,1))
abline(v=age.x, lty=2, col=c(2:4,1), lwd=2)
legend( 80, 0.01125, c("Total", "Free light fraction", "Occluded light fraction", "Dense fraction"), col=1:
#text(x=age.x, y=age.y, pos=c(4,4,4,2), labels=b.age.text, col=c(2:4,1))
dev.off()
}

#load data from model fits
input.rds <- readRDS(paste0(mainDir, dataDir, "/R_mcmc_output.rds"))

#bring in dataset output from input test
In.t <- read.xlsx(paste0(mainDir,dataDir, "/R_output_modelpars_input_tests.xlsx"),
                  sheet = 1)

#get all mcmc.mod outputs
input.rds.mcmc <- lapply(input.rds, `[[`, 1)

#subset data to 1.8 C inputs and graph
i.c <- 2.50
s.row <- which(round(In.t$Cinput,2) == i.c)
p_1.8 <- data.frame(In.t[s.row,c(2:8,1)])
p_mcmc_1.8 <- input.rds.mcmc[[s.row]]
graph.pars(p_mcmc_1.8, p_1.8, pt, fractions, pt.data)
```

```
## pdf
##   2
```

```
figDir <- "/Figures/Final publication figures"
knitr::include_graphics(paste0(mainDir,figDir, "/fractions_", i.c, "MgC_input.pdf"))
```




```
knitr::include_graphics(paste0(mainDir,figDir, "/TransitTime_", i.c, "MgC_input.pdf"))
```




### 4.2 median transit times

I didn't pull the median transit time from the models ran previously across a range of C inputs. I already ran the code across inputs and saved the mcmc runs to an RDS file and the best model parameters (and pool ages/transit times) to an excel sheet. To avoid re-running and potentially changing the output (changes would be very small but mcmc solves slightly differently each time) that we already added to tables and graphs, and considering the systemAge() and transitTime() functions only require the input and model parameters, I simply calculate them here after the fact.

```
# bring in the input value used and best parameters from mcmc models
In.t <- read.xlsx(paste0(mainDir,dataDir, "/R_output_modelpars_input_tests.xlsx"),
                  sheet = 1)

#subset to range of inputs used in paper (2 - 3 MgC/ha/yr)
m.t <- In.t[In.t$Cinput >= 2.0 & In.t$Cinput <= 3.0, c(1:8)]
```

```r
#function to apply across dataframe
median.transit <- function(m.t) {

  Cinput <- m.t[1]
  b.b.par <- m.t[-1]
  #input matrix
  In <- matrix(c(Cinput,0,0), ncol=1)
  #make bayesian A matrix
  b.A <- diag(-b.b.par[1:3])
  b.A[2,1]=b.b.par[4]*b.b.par[1]
  b.A[1,2]=b.b.par[5]*b.b.par[2]
  b.A[3,2]=b.b.par[6]*b.b.par[2]
  b.A[2,3]=b.b.par[7]*b.b.par[3]
  # sequence of number for the computation of the distribution
  tau=seq(0,3000, by=0.5)
  #calculate the system age and transit time for classical and bayesian solutions
  b.ages=systemAge(A=b.A, u=In, a=tau, q=c(0.5))
  b.transitTimes=transitTime(A=b.A, u=In, a=tau, q=c(0.5))
  #take just mean/median ages
  ages <- cbind(b.ages$meanSystemAge,
                b.ages$quantilesSystemAge,
                b.transitTimes$meanTransitTime,
                b.transitTimes$quantiles,
                b.ages$meanPoolAge[1],
                b.ages$meanPoolAge[2],
                b.ages$meanPoolAge[3])
  return(ages)
}
#apply function across rows of dataframe containing models
m.med <- data.frame(t(apply(m.t, 1, median.transit)))

#bind back to Cinputs and model parameters
mean.median <- cbind(m.t, m.med)
colnames(mean.median)[9:15] <- c("Mean system age", "Median system age",
                                 "Mean transit time", "Median transit time",
                                 "Mean p1 age", "Mean p2 age", "Mean p3 age")

#table of mean and sd for all parameters and mean ages
mean.m <- cbind(apply(mean.median, 2, mean), apply(mean.median, 2, sd))
colnames(mean.m)[1:2] <- c("mean", "sd")
round(mean.m, digits=3) %>%
  kable(booktabs=TRUE) %>%
    kable_styling(c("striped"))
```

| | mean | sd |
|---|---|---|
| Cinput | 2.500 | 0.310 |
| k1 | 1.171 | 0.129 |
| k2 | 0.537 | 0.034 |
| k3 | 0.010 | 0.002 |
| alpha21 | 0.897 | 0.013 |
| alpha12 | 0.145 | 0.019 |
| alpha32 | 0.812 | 0.023 |
| alpha23 | 0.545 | 0.022 |
| Mean system age | 206.308 | 43.579 |
| Median system age | 142.910 | 30.210 |
| Mean transit time | 176.522 | 39.311 |
| Median transit time | 110.344 | 25.976 |
| Mean p1 age | 26.454 | 6.444 |
| Mean p2 age | 109.737 | 24.107 |
| Mean p3 age | 209.711 | 43.709 |