BIDS project: Soil

Samuel Froehlich, Heiko Holziger, Fabian Meyer, Michael Zimmermann

May 27, 2020

Contents


```r
library('tidyverse')
```

```
## -- Attaching packages ----------------------------------------------------------------------------

## v ggplot2 3.2.1     v purrr   0.3.3
```

```
## v tibble  2.1.3     v dplyr   0.8.4
## v tidyr   1.0.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0

## -- Conflicts ------------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library('cluster')
library('factoextra')
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```r
library('shiny')
library('ggloop')
library('ggpubr')
library('parsnip')
library('tidymodels')
```

```
## Registered S3 methods overwritten by 'lme4':
##   method                          from
##   cooks.distance.influence.merMod car
##   influence.merMod                car
##   dfbeta.influence.merMod         car
##   dfbetas.influence.merMod        car

## -- Attaching packages ---------------------------------------------------------------------------

## v broom     0.5.4      v rsample   0.0.6
## v dials     0.0.6      v tune      0.1.0
## v infer     0.5.1      v workflows 0.1.1
## v recipes   0.1.12     v yardstick 0.0.6

## -- Conflicts ------------------------------------------------------------------------------------
## x scales::discard() masks purrr::discard()
## x dplyr::filter()   masks stats::filter()
## x recipes::fixed()  masks stringr::fixed()
## x dplyr::lag()      masks stats::lag()
## x dials::margin()   masks ggplot2::margin()
## x yardstick::spec() masks readr::spec()
## x recipes::step()   masks stats::step()
```

```r
library('broom')
library('yardstick')
library('knitr')
library("rmarkdown")
library('readxl')
theme_set(theme_pubr())
```

1 Load the data set soil.csv and view it.

```r
soil <- read_csv('soil.csv')
```

```
## Parsed with column specification:
## cols(
##   Clay1 = col_double(),
##   Clay2 = col_double(),
```

```
##   Clay5 = col_double(),
##   CEC1 = col_double(),
##   CEC2 = col_double(),
##   CEC5 = col_double(),
##   OC1 = col_double(),
##   OC2 = col_double(),
##   OC5 = col_double()
## )
```

2 Discuss the dataset based on str() and summary().

```r
dim(soil)
```

```
## [1] 147   9
```

```r
str(soil)
```

```
## Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame': 147 obs. of  9 variables:
##  $ Clay1: num  72 71 61 55 47 49 63 59 46 62 ...
##  $ Clay2: num  74 75 59 62 56 53 66 66 56 63 ...
##  $ Clay5: num  78 80 66 61 53 57 70 72 70 62 ...
##  $ CEC1 : num  13.6 12.6 21.7 11.6 14.9 18.2 14.9 14.6 7.9 14.9 ...
##  $ CEC2 : num  10.1 8.2 10.2 8.4 9.2 11.6 7.4 7.1 5.7 6.8 ...
##  $ CEC5 : num  7.1 7.4 6.6 8 8.5 6.2 5.4 7 4.5 6 ...
##  $ OC1  : num  5.5 3.2 6.98 3.19 4.4 5.31 4.55 4.5 2.3 7.34 ...
##  $ OC2  : num  3.1 1.7 2.4 1.5 1.2 3.2 2.15 1.42 1.36 2.54 ...
##  $ OC5  : num  1.5 1 1.3 1.26 0.8 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Clay1 = col_double(),
##   ..   Clay2 = col_double(),
##   ..   Clay5 = col_double(),
##   ..   CEC1 = col_double(),
##   ..   CEC2 = col_double(),
##   ..   CEC5 = col_double(),
##   ..   OC1 = col_double(),
##   ..   OC2 = col_double(),
##   ..   OC5 = col_double()
##   .. )
```

```r
summary(soil)
```

```
##      Clay1           Clay2           Clay5            CEC1            CEC2      
##  Min.   :10.00   Min.   : 8.00   Min.   :16.00   Min.   : 3.0   Min.   : 1.60  
##  1st Qu.:21.00   1st Qu.:27.00   1st Qu.:36.50   1st Qu.: 7.5   1st Qu.: 5.00  
##  Median :30.00   Median :36.00   Median :44.00   Median :10.1   Median : 7.00  
##  Mean   :31.27   Mean   :36.75   Mean   :44.68   Mean   :11.2   Mean   : 7.41  
##  3rd Qu.:39.00   3rd Qu.:47.00   3rd Qu.:54.00   3rd Qu.:13.1   3rd Qu.: 9.40  
##  Max.   :72.00   Max.   :75.00   Max.   :80.00   Max.   :29.0   Max.   :22.00  
##       CEC5             OC1              OC2              OC5        
##  Min.   : 1.000   Min.   : 1.040   Min.   :0.300   Min.   :0.2000  
##  1st Qu.: 5.000   1st Qu.: 1.975   1st Qu.:0.850   1st Qu.:0.6000  
##  Median : 6.500   Median : 2.700   Median :1.300   Median :0.8400  
##  Mean   : 6.844   Mean   : 2.987   Mean   :1.386   Mean   :0.8103  
##  3rd Qu.: 8.900   3rd Qu.: 3.700   3rd Qu.:1.700   3rd Qu.:1.0000  
```

```
##  Max.   :14.000   Max.   :10.900   Max.   :3.700   Max.   :1.7000
```

```
colnames(soil)
```

```
## [1] "Clay1" "Clay2" "Clay5" "CEC1"  "CEC2"  "CEC5"  "OC1"   "OC2"   "OC5"
```

```
typeof(soil)
```

```
## [1] "list"
```

The head() and tail() functions default to 6 rows, but we can adjust the number of rows using the "n =" argument

```
head(soil, n = 10)
```

```
## # A tibble: 10 x 9
##    Clay1 Clay2 Clay5 CEC1  CEC2  CEC5   OC1   OC2   OC5
##    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1    72    74    78  13.6  10.1   7.1  5.5   3.1   1.5
##  2    71    75    80  12.6   8.2   7.4  3.2   1.7   1
##  3    61    59    66  21.7  10.2   6.6  6.98  2.4   1.3
##  4    55    62    61  11.6   8.4   8    3.19  1.5   1.26
##  5    47    56    53  14.9   9.2   8.5  4.4   1.2   0.8
##  6    49    53    57  18.2  11.6   6.2  5.31  3.2   1.08
##  7    63    66    70  14.9   7.4   5.4  4.55  2.15  1.23
##  8    59    66    72  14.6   7.1   7    4.5   1.42  1.3
##  9    46    56    70   7.9   5.7   4.5  2.3   1.36  0.9
## 10    62    63    62  14.9   6.8   6    7.34  2.54  1.7
```

```
tail(soil, n = 10)
```

```
## # A tibble: 10 x 9
##    Clay1 Clay2 Clay5 CEC1  CEC2  CEC5   OC1   OC2   OC5
##    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1    21    41    47  8     8     9     1.7   0.9   0.8
##  2    42    60    66  9     8     8     2.3   1.2   1
##  3    19    21    38 15    13    11     1.23  0.82  0.74
##  4    33    36    40 13    13    10     2.5   2.2   1.3
##  5    45    50    57 10     8.3   8.3  4.2   1.9   1.1
##  6    36    46    47 13    12     9     3.1   1.4   1
##  7    25    38    39  6     5     5     1.5   0.8   0.8
##  8    30    18    23  7     6     7     1.5   0.8   0.8
##  9    34    40    45 13.2  12.2  11.7  3.6   2     1
## 10    30    38    46  6.9   4.7   2.9  2.7   1.6   0.75
```

While the first 6 functions are printed to the console, the View() function opens a table in another window

```
View(soil)
```

Convert to tidyverse

```
soil_tibble <- as_tibble(soil)
View(soil_tibble)
typeof(soil_tibble)
```

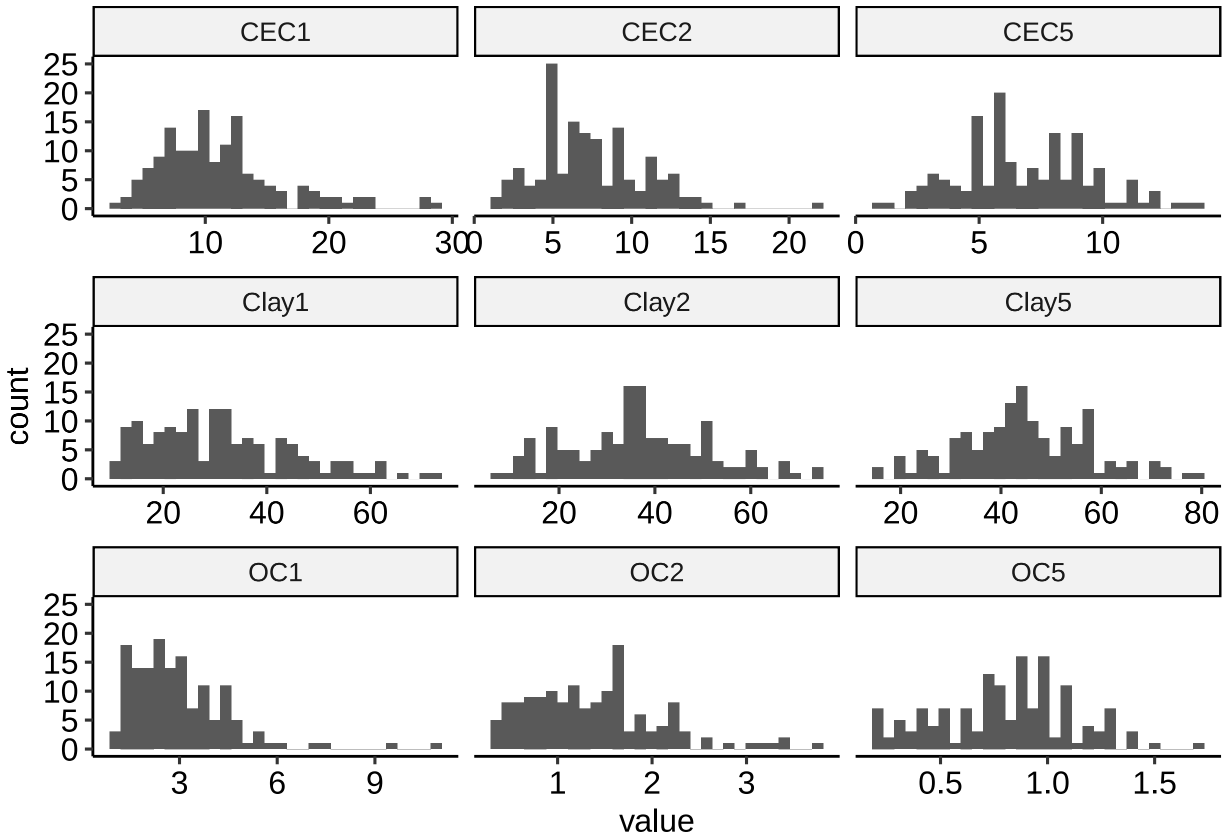
```
## [1] "list"
```

```
is_tibble(soil_tibble)
```

```
## [1] TRUE
```

Make plots for each row to get an overview

```
ggplot(gather(soil_tibble), aes(value)) +
  geom_histogram(bins = 30) +
  facet_wrap(~key, scales = 'free_x')
```



2.1 Remarks

- Some of the data (CEC1, CEC2, Clay1, OC1, OC2) is left-skewed.
- Mostly in the first layer (1).
- The range of Clay on all three levels is between 8.0 and 80. The lower you dig the higher the clay percentage is.
- OC: The Range is between 1.6 and 29.0. The Soil in the first 10 cm is the most potent. The lower you dig the worsen the soil gets.
- CEC: The Range is between 0.2 and 10.9. The top-level soil has a higher mean than lower level.

3 Your goal is to use Clay and OC to predict CEC. Get a first impression on the predictive capabilities of your data by plotting all the Clay and OC variables against all CEC variables. Discuss your plots.

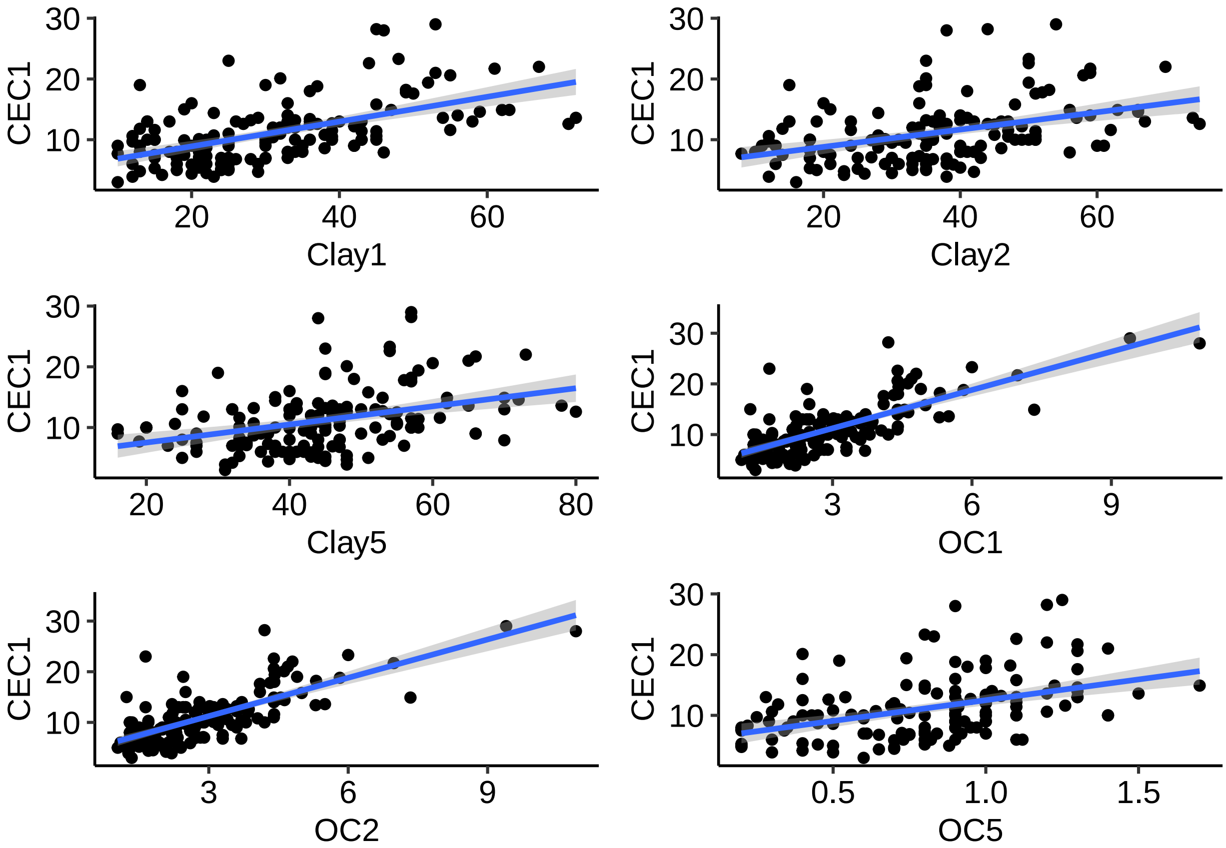
Plot every Clay and OC against every CEC (18 different combinations)

```
plots <- list()
plots$plot1 <- ggplot(data=soil_tibble, aes(x=soil_tibble$Clay1, y=soil_tibble$CEC1)) +
```

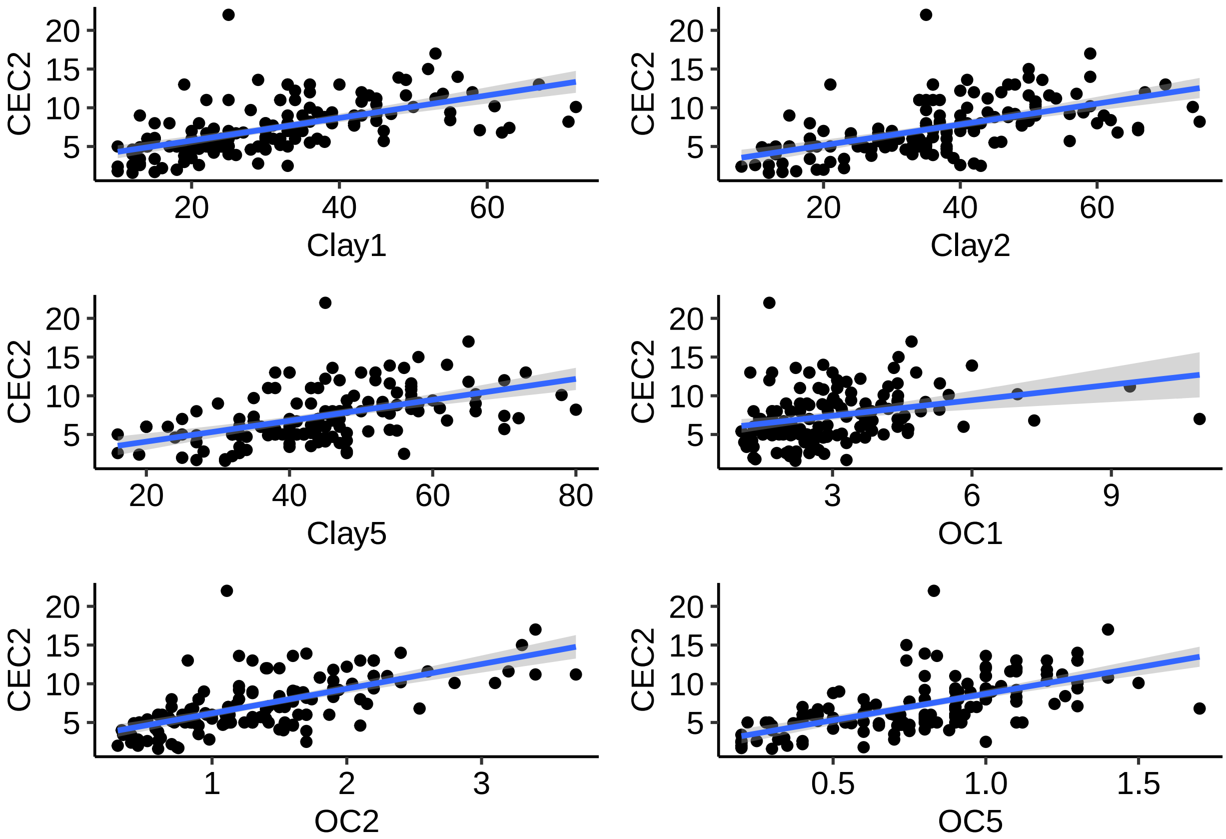
```
    geom_point() + geom_smooth(method = 'lm') + xlab('Clay1') + ylab('CEC1')
plots$plot2 <- ggplot(data=soil_tibble, aes(x=soil_tibble$Clay2, y=soil_tibble$CEC1)) +
    geom_point() + geom_smooth(method = 'lm') + xlab('Clay2') + ylab('CEC1')
plots$plot3 <- ggplot(data=soil_tibble, aes(x=soil_tibble$Clay5, y=soil_tibble$CEC1)) +
    geom_point() + geom_smooth(method = 'lm') + xlab('Clay5') + ylab('CEC1')
plots$plot4 <- ggplot(data=soil_tibble, aes(x=soil_tibble$OC1, y=soil_tibble$CEC1)) +
    geom_point() + geom_smooth(method = 'lm') + xlab('OC1') + ylab('CEC1')
plots$plot5 <- ggplot(data=soil_tibble, aes(x=soil_tibble$OC1, y=soil_tibble$CEC1)) +
  geom_point() + geom_smooth(method = 'lm') + xlab('OC2') + ylab('CEC1')
plots$plot6 <- ggplot(data=soil_tibble, aes(x=soil_tibble$OC5, y=soil_tibble$CEC1)) +
  geom_point() + geom_smooth(method = 'lm') + xlab('OC5') + ylab('CEC1')
plots$plot7 <- ggplot(data=soil_tibble, aes(x=soil_tibble$Clay1, y=soil_tibble$CEC2)) +
  geom_point() + geom_smooth(method = 'lm') + xlab('Clay1') + ylab('CEC2')
plots$plot8 <- ggplot(data=soil_tibble, aes(x=soil_tibble$Clay2, y=soil_tibble$CEC2)) +
  geom_point() + geom_smooth(method = 'lm') + xlab('Clay2') + ylab('CEC2')
plots$plot9 <- ggplot(data=soil_tibble, aes(x=soil_tibble$Clay5, y=soil_tibble$CEC2)) +
  geom_point() + geom_smooth(method = 'lm') + xlab('Clay5') + ylab('CEC2')
plots$plot10 <- ggplot(data=soil_tibble, aes(x=soil_tibble$OC1, y=soil_tibble$CEC2)) +
  geom_point() + geom_smooth(method = 'lm') + xlab('OC1') + ylab('CEC2')
plots$plot11 <- ggplot(data=soil_tibble, aes(x=soil_tibble$OC2, y=soil_tibble$CEC2)) +
  geom_point() + geom_smooth(method = 'lm') + xlab('OC2') + ylab('CEC2')
plots$plot12 <- ggplot(data=soil_tibble, aes(x=soil_tibble$OC5, y=soil_tibble$CEC2)) +
  geom_point() + geom_smooth(method = 'lm') + xlab('OC5') + ylab('CEC2')
plots$plot13 <- ggplot(data=soil_tibble, aes(x=soil_tibble$Clay1, y=soil_tibble$CEC5)) +
  geom_point() + geom_smooth(method = 'lm') + xlab('Clay1') + ylab('CEC5')
plots$plot14 <- ggplot(data=soil_tibble, aes(x=soil_tibble$Clay2, y=soil_tibble$CEC5)) +
  geom_point() + geom_smooth(method = 'lm') + xlab('Clay2') + ylab('CEC5')
plots$plot15 <- ggplot(data=soil_tibble, aes(x=soil_tibble$Clay5, y=soil_tibble$CEC5)) +
  geom_point() + geom_smooth(method = 'lm') + xlab('Clay5') + ylab('CEC5')
plots$plot16 <- ggplot(data=soil_tibble, aes(x=soil_tibble$OC1, y=soil_tibble$CEC5)) +
  geom_point() + geom_smooth(method = 'lm') + xlab('OC1') + ylab('CEC5')
plots$plot17 <- ggplot(data=soil_tibble, aes(x=soil_tibble$OC2, y=soil_tibble$CEC5)) +
  geom_point() + geom_smooth(method = 'lm') + xlab('OC2') + ylab('CEC5')
plots$plot18 <- ggplot(data=soil_tibble, aes(x=soil_tibble$OC5, y=soil_tibble$CEC5)) +
  geom_point() + geom_smooth(method = 'lm') + xlab('OC5') + ylab('CEC5')
```

Put dependent variable on one single plot

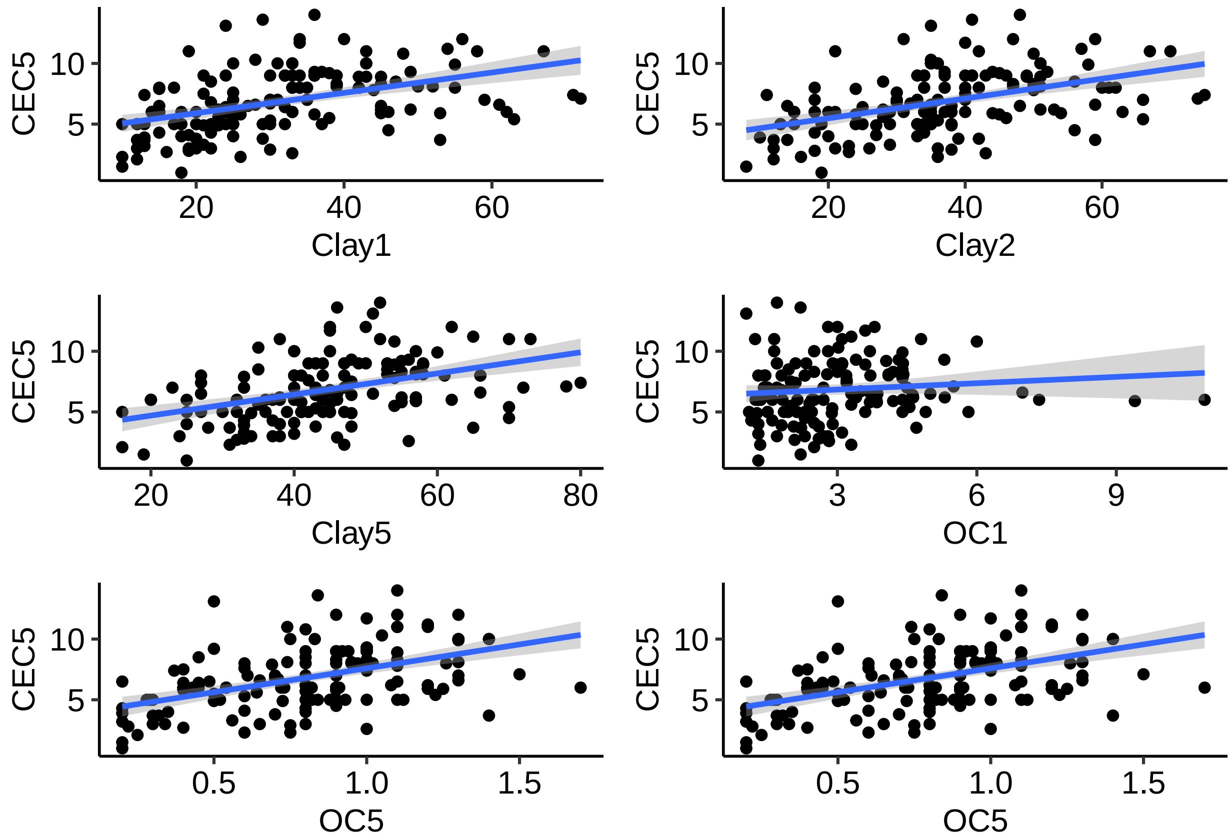
```
figures <- list()
figures$figure1 <- ggarrange(plots$plot1, plots$plot2, plots$plot3, plots$plot4, plots$plot5, plots$plo
                     ncol = 2, nrow = 3)
figures$figure1
```




```r
figures$figure2 <- ggarrange(plots$plot7, plots$plot8, plots$plot9, plots$plot10, plots$plot11, plots$pl
                             ncol = 2, nrow = 3)
figures$figure2
```




```
figures$figure3 <- ggarrange(plots$plot13, plots$plot14, plots$plot15, plots$plot16, plots$plot18, plots
                   ncol = 2, nrow = 3)
figures$figure3
```



3.1 Remarks

- CEC1: We clearly can see that OC1 and OC2 have the biggest influence on CEC1.
- CEC1: Clay5 and OC5 have the mildest curve.
- CEC1: OC1 on CEC1 is very left-skewed.
- CEC2: OC2 is best for CEC2.
- CEC2: OC1 on CEC2 is left-skewed.
- CEC5: A middle dose of OC5 (between 1.0 and 1.5) is best for cec5. it has the steepest curve.
- CEC5: Clay1, Clay2 and Clay3 look good too
- CEC5: OC1 is very left-skewed data and looks it doesn't have a big effect.
- All in all it can be said that predictors from within the same layer have the biggest influence on CEC values.

4 Build simple linear regression models for all of the above variable pairs. Plot your results.

```
simple_linear_models <- list()

simple_linear_models$lm1 <- lm(CEC1 ~ Clay1, data = soil_tibble)
simple_linear_models$sum1 <- summary(simple_linear_models$lm1);
simple_linear_models$sum1
```

```
##
## Call:
## lm(formula = CEC1 ~ Clay1, data = soil_tibble)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.7065 -3.3512 -0.6446  2.2007 14.1962
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.82623    0.86195   5.599 1.05e-07 ***
## Clay1        0.20395    0.02519   8.096 2.11e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.242 on 145 degrees of freedom
## Multiple R-squared:  0.3113, Adjusted R-squared:  0.3066
## F-statistic: 65.55 on 1 and 145 DF,  p-value: 2.107e-13
```

```r
simple_linear_models$lm2 <- lm(CEC1 ~ Clay2, data = soil_tibble)
simple_linear_models$sum2 <- summary(simple_linear_models$lm2);
simple_linear_models$sum2
```

```
##
## Call:
## lm(formula = CEC1 ~ Clay2, data = soil_tibble)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.4825 -3.5900 -0.5548  1.9026 16.6175
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.96482    1.04324   5.718 5.95e-08 ***
## Clay2        0.14257    0.02639   5.403 2.63e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.664 on 145 degrees of freedom
## Multiple R-squared:  0.1676, Adjusted R-squared:  0.1618
## F-statistic: 29.19 on 1 and 145 DF,  p-value: 2.629e-07
```

```r
simple_linear_models$lm3 <- lm(CEC1 ~ Clay5, data = soil_tibble)
simple_linear_models$sum3 <- summary(simple_linear_models$lm3);
simple_linear_models$sum3
```

```
##
## Call:
## lm(formula = CEC1 ~ Clay5, data = soil_tibble)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.7993 -3.1993 -0.9093  2.4773 16.8974
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.53939    1.41561   3.207  0.00165 **
## Clay5        0.14916    0.03045   4.898 2.55e-06 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.735 on 145 degrees of freedom
## Multiple R-squared:  0.142,  Adjusted R-squared:  0.136
## F-statistic: 23.99 on 1 and 145 DF,  p-value: 2.554e-06
```

```r
simple_linear_models$lm4 <- lm(CEC1 ~ OC1, data = soil_tibble)
simple_linear_models$sum4 <- summary(simple_linear_models$lm4);
simple_linear_models$sum4
```

```
##
## Call:
## lm(formula = CEC1 ~ OC1, data = soil_tibble)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.2809 -2.2469 -0.2099  1.5770 15.1936
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.6706     0.6304   5.823 3.57e-08 ***
## OC1           2.5218     0.1887  13.365  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.421 on 145 degrees of freedom
## Multiple R-squared:  0.552,  Adjusted R-squared:  0.5489
## F-statistic: 178.6 on 1 and 145 DF,  p-value: < 2.2e-16
```

```r
simple_linear_models$lm5 <- lm(CEC1 ~ OC2, data = soil_tibble)
simple_linear_models$sum5 <- summary(simple_linear_models$lm5);
simple_linear_models$sum5
```

```
##
## Call:
## lm(formula = CEC1 ~ OC2, data = soil_tibble)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.838 -2.794 -0.745  1.944 16.294
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5.1255     0.7552   6.787 2.72e-10 ***
## OC2           4.3872     0.4875   8.999 1.17e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.094 on 145 degrees of freedom
## Multiple R-squared:  0.3584, Adjusted R-squared:  0.3539
## F-statistic: 80.98 on 1 and 145 DF,  p-value: 1.167e-15
```

```r
simple_linear_models$lm6 <- lm(CEC1 ~ OC5, data = soil_tibble)
simple_linear_models$sum6 <- summary(simple_linear_models$lm6);
simple_linear_models$sum6
```

```
##
## Call:
## lm(formula = CEC1 ~ OC5, data = soil_tibble)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.3188 -3.2873 -0.4848  1.5734 16.1834
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    5.671      1.040   5.454 2.07e-07 ***
## OC5            6.828      1.194   5.719 5.91e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.617 on 145 degrees of freedom
## Multiple R-squared:  0.184,  Adjusted R-squared:  0.1784
## F-statistic: 32.71 on 1 and 145 DF,  p-value: 5.91e-08
```

```
simple_linear_models$lm7 <- lm(CEC2 ~ Clay1, data = soil_tibble)
simple_linear_models$sum7 <- summary(simple_linear_models$lm7);
simple_linear_models$sum7
```

```
##
## Call:
## lm(formula = CEC2 ~ Clay1, data = soil_tibble)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.1620 -1.8148 -0.3505  1.0694 15.5038
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.85314    0.57068   5.000 1.63e-06 ***
## Clay1        0.14572    0.01668   8.737 5.35e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.808 on 145 degrees of freedom
## Multiple R-squared:  0.3449, Adjusted R-squared:  0.3404
## F-statistic: 76.34 on 1 and 145 DF,  p-value: 5.347e-15
```

```
simple_linear_models$lm8 <- lm(CEC2 ~ Clay2, data = soil_tibble)
simple_linear_models$sum8 <- summary(simple_linear_models$lm8);
simple_linear_models$sum8
```

```
##
## Call:
## lm(formula = CEC2 ~ Clay2, data = soil_tibble)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.7494 -1.8413 -0.3098  1.1066 14.8245
##
## Coefficients:
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.47707    0.63895   3.877  0.00016 ***
## Clay2        0.13424    0.01616   8.306  6.4e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.856 on 145 degrees of freedom
## Multiple R-squared:  0.3224, Adjusted R-squared:  0.3177
## F-statistic: 68.99 on 1 and 145 DF,  p-value: 6.401e-14
```

```
simple_linear_models$lm9 <- lm(CEC2 ~ Clay5, data = soil_tibble)
simple_linear_models$sum9 <- summary(simple_linear_models$lm9);
simple_linear_models$sum9
```

```
##
## Call:
## lm(formula = CEC2 ~ Clay5, data = soil_tibble)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.4376 -1.9311 -0.3184  1.6877 14.5467
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.3814     0.8972   1.540    0.126
## Clay5         0.1349     0.0193   6.991 9.23e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.001 on 145 degrees of freedom
## Multiple R-squared:  0.2521, Adjusted R-squared:  0.2469
## F-statistic: 48.87 on 1 and 145 DF,  p-value: 9.232e-11
```

```
simple_linear_models$lm10 <- lm(CEC2 ~ OC1, data = soil_tibble)
simple_linear_models$sum10 <- summary(simple_linear_models$lm10);
simple_linear_models$sum10
```

```
##
## Call:
## lm(formula = CEC2 ~ OC1, data = soil_tibble)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.9194 -2.3052 -0.5001  1.7147 15.4911
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5.4118     0.6117   8.847 2.84e-15 ***
## OC1           0.6690     0.1831   3.653 0.000361 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.32 on 145 degrees of freedom
## Multiple R-squared:  0.08429,    Adjusted R-squared:  0.07797
## F-statistic: 13.35 on 1 and 145 DF,  p-value: 0.000361
```

```r
simple_linear_models$lm11 <- lm(CEC2 ~ OC2, data = soil_tibble)
simple_linear_models$sum11 <- summary(simple_linear_models$lm11);
simple_linear_models$sum11
```

```
##
## Call:
## lm(formula = CEC2 ~ OC2, data = soil_tibble)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.9103 -1.7105 -0.2103  1.0162 15.4660
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.0039     0.4922   6.104 9.01e-09 ***
## OC2           3.1802     0.3177  10.010  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.668 on 145 degrees of freedom
## Multiple R-squared:  0.4086, Adjusted R-squared:  0.4045
## F-statistic: 100.2 on 1 and 145 DF,  p-value: < 2.2e-16
```

```r
simple_linear_models$lm12 <- lm(CEC2 ~ OC5, data = soil_tibble)
simple_linear_models$sum12 <- summary(simple_linear_models$lm12);
simple_linear_models$sum12
```

```
##
## Call:
## lm(formula = CEC2 ~ OC5, data = soil_tibble)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.6883 -1.7022 -0.5061  1.2003 14.4553
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.8745     0.6054   3.096  0.00235 **
## OC5           6.8316     0.6952   9.827  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.688 on 145 degrees of freedom
## Multiple R-squared:  0.3998, Adjusted R-squared:  0.3956
## F-statistic: 96.57 on 1 and 145 DF,  p-value: < 2.2e-16
```

```r
simple_linear_models$lm13 <- lm(CEC5 ~ Clay1, data = soil_tibble)
simple_linear_models$sum13 <- summary(simple_linear_models$lm13);
simple_linear_models$sum13
```

```
##
## Call:
## lm(formula = CEC5 ~ Clay1, data = soil_tibble)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -4.9636 -1.7247 -0.0377  1.5123  6.9460
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.22563    0.47705   8.858 2.66e-15 ***
## Clay1        0.08374    0.01394   6.006 1.46e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.348 on 145 degrees of freedom
## Multiple R-squared:  0.1992, Adjusted R-squared:  0.1937
## F-statistic: 36.07 on 1 and 145 DF,  p-value: 1.46e-08
```

```r
simple_linear_models$lm14 <- lm(CEC5 ~ Clay2, data = soil_tibble)
simple_linear_models$sum14 <- summary(simple_linear_models$lm14);
simple_linear_models$sum14
```

```
##
## Call:
## lm(formula = CEC5 ~ Clay2, data = soil_tibble)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.9573 -1.7018  0.0686  1.4242  6.4094
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.84999    0.52236   7.370 1.19e-11 ***
## Clay2        0.08148    0.01321   6.167 6.58e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.335 on 145 degrees of freedom
## Multiple R-squared:  0.2078, Adjusted R-squared:  0.2023
## F-statistic: 38.03 on 1 and 145 DF,  p-value: 6.578e-09
```

```r
simple_linear_models$lm15 <- lm(CEC5 ~ Clay5, data = soil_tibble)
simple_linear_models$sum15 <- summary(simple_linear_models$lm15);
simple_linear_models$sum15
```

```
##
## Call:
## lm(formula = CEC5 ~ Clay5, data = soil_tibble)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.2271 -1.8720 -0.1354  1.5622  6.6412
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.96458    0.70915   4.180 5.01e-05 ***
## Clay5        0.08683    0.01526   5.692 6.73e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 2.372 on 145 degrees of freedom
## Multiple R-squared:  0.1826, Adjusted R-squared:  0.177
## F-statistic:  32.4 on 1 and 145 DF,  p-value: 6.734e-08
```

```
simple_linear_models$lm16 <- lm(CEC5 ~ OC1, data = soil_tibble)
simple_linear_models$sum16 <- summary(simple_linear_models$lm16);
simple_linear_models$sum16
```

```
##
## Call:
## lm(formula = CEC5 ~ OC1, data = soil_tibble)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.5514 -1.7293 -0.4957  1.9292  7.3792
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.3258     0.4809  13.153   <2e-16 ***
## OC1           0.1735     0.1440   1.205     0.23
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.61 on 145 degrees of freedom
## Multiple R-squared:  0.009921,   Adjusted R-squared:  0.003093
## F-statistic: 1.453 on 1 and 145 DF,  p-value: 0.23
```

```
simple_linear_models$lm17 <- lm(CEC5 ~ OC2, data = soil_tibble)
simple_linear_models$sum17 <- summary(simple_linear_models$lm17);
simple_linear_models$sum17
```

```
##
## Call:
## lm(formula = CEC5 ~ OC2, data = soil_tibble)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -5.368 -1.844 -0.242  1.755  7.250
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5.3147     0.4626  11.489  < 2e-16 ***
## OC2           1.1040     0.2986   3.697 0.000309 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.508 on 145 degrees of freedom
## Multiple R-squared:  0.08613,    Adjusted R-squared:  0.07983
## F-statistic: 13.67 on 1 and 145 DF,  p-value: 0.0003089
```

```
simple_linear_models$lm18 <- lm(CEC5 ~ OC5, data = soil_tibble)
simple_linear_models$sum18 <- summary(simple_linear_models$lm18);
simple_linear_models$sum18
```

```
##
```

```
## Call:
## lm(formula = CEC5 ~ OC5, data = soil_tibble)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.4645 -1.7873  0.2194  1.4094  7.4768
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.6559     0.5178   7.061 6.35e-11 ***
## OC5           3.9347     0.5946   6.618 6.58e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.299 on 145 degrees of freedom
## Multiple R-squared:  0.232,  Adjusted R-squared:  0.2267
## F-statistic:  43.8 on 1 and 145 DF,  p-value: 6.579e-10
```

5 Based on the R-squared value, what is the best predictor for top-soil CEC (CEC1), mid-soil CEC (CEC2) and sub-soil CEC (CEC5), respectively?

```r
rsq_values <- list(
  simple_linear_models$sum1$r.squared,
  simple_linear_models$sum2$r.squared,
  simple_linear_models$sum3$r.squared,
  simple_linear_models$sum4$r.squared,
  simple_linear_models$sum5$r.squared,
  simple_linear_models$sum6$r.squared,
  simple_linear_models$sum7$r.squared,
  simple_linear_models$sum8$r.squared,
  simple_linear_models$sum9$r.squared,
  simple_linear_models$sum10$r.squared,
  simple_linear_models$sum11$r.squared,
  simple_linear_models$sum12$r.squared,
  simple_linear_models$sum13$r.squared,
  simple_linear_models$sum14$r.squared,
  simple_linear_models$sum15$r.squared,
  simple_linear_models$sum16$r.squared,
  simple_linear_models$sum17$r.squared,
  simple_linear_models$sum18$r.squared
)
```

Take rsq_values list and convert to tibble

```r
rsq_values <- as.data.frame(rsq_values)
rsq_values <- t(rsq_values)
rsq_values <- as.data.frame(rsq_values)
rsq_values_tibble <- as_tibble(rsq_values)
```

Rename column

```
rsq_values_tibble <- rsq_values_tibble %>% rename(r_squared = V1)
```

Add name column

```
rsq_values_tibble <- rsq_values_tibble %>% add_column(name = 1:18)
```

Reorder data.frame / tibble

```
rsq_values_tibble <- rsq_values_tibble[c('name', 'r_squared')]
```

Order tibble by descending R^2 value

```
rsq_values_tibble %>% arrange(desc(r_squared))
```

```
## # A tibble: 18 x 2
##     name r_squared
##    <int>     <dbl>
##  1     4   0.552
##  2    11   0.409
##  3    12   0.400
##  4     5   0.358
##  5     7   0.345
##  6     8   0.322
##  7     1   0.311
##  8     9   0.252
##  9    18   0.232
## 10    14   0.208
## 11    13   0.199
## 12     6   0.184
## 13    15   0.183
## 14     2   0.168
## 15     3   0.142
## 16    17   0.0861
## 17    10   0.0843
## 18    16   0.00992
```

5.1 Remarks

- Best: Model 4 (CEC1 ~ OC1)
- Second: model 11 (CEC2 ~ OC2)
- Third: model 12 (CEC2 ~ OC5)
- Fourth: model 5 (CEC1 ~ OC2)
- Best for CEC5 is on rank 9: model 18 (CEC5 ~ OC5)
- Interpretation: In general OC levels are much more important than clay levels
- This means there is a stronger positive correlation between CEC and OC levels than between CEC and Clay levels
- The most important OC level is the one from the same depth of the CEC / soil itself
- The second most important is the one that is beneath
- This suggests that organic carbon is able to move upwards in soil

Question: If we compare p-values, do these values correspond to the R^2-values?

Function to extract the overall ANOVA p-value out of a linear model object summary

```
lmp <- function(model_summary) {
  if (class(model_summary) != "summary.lm") stop("Not an object of class 'lm' ")
  f <- model_summary$fstatistic
```

```
  p <- pf(f[1],f[2],f[3],lower.tail=F)
  attributes(p) <- NULL
  return(p)
}
```

```
p_values <- list(
  lmp(simple_linear_models$sum1),
  lmp(simple_linear_models$sum2),
  lmp(simple_linear_models$sum3),
  lmp(simple_linear_models$sum4),
  lmp(simple_linear_models$sum5),
  lmp(simple_linear_models$sum6),
  lmp(simple_linear_models$sum7),
  lmp(simple_linear_models$sum8),
  lmp(simple_linear_models$sum9),
  lmp(simple_linear_models$sum10),
  lmp(simple_linear_models$sum11),
  lmp(simple_linear_models$sum12),
  lmp(simple_linear_models$sum13),
  lmp(simple_linear_models$sum14),
  lmp(simple_linear_models$sum15),
  lmp(simple_linear_models$sum16),
  lmp(simple_linear_models$sum17),
  lmp(simple_linear_models$sum18)
)
```

Take p_values list and convert to tibble

```
p_values <- as.data.frame(p_values)
p_values <- t(p_values)
p_values <- as.data.frame(p_values)
p_values_tibble <- as_tibble(p_values)
```

Rename column

```
names(p_values_tibble)[1] <- 'p_value'
```

Add name column

```
p_values_tibble <- p_values_tibble %>% add_column(name = 1:18)
```

Reorder data.frame / tibble

```
p_values_tibble <- p_values_tibble[c('name', 'p_value')]
```

Order tibble by ascending p-value

```
p_values_tibble %>% arrange(p_value)
```

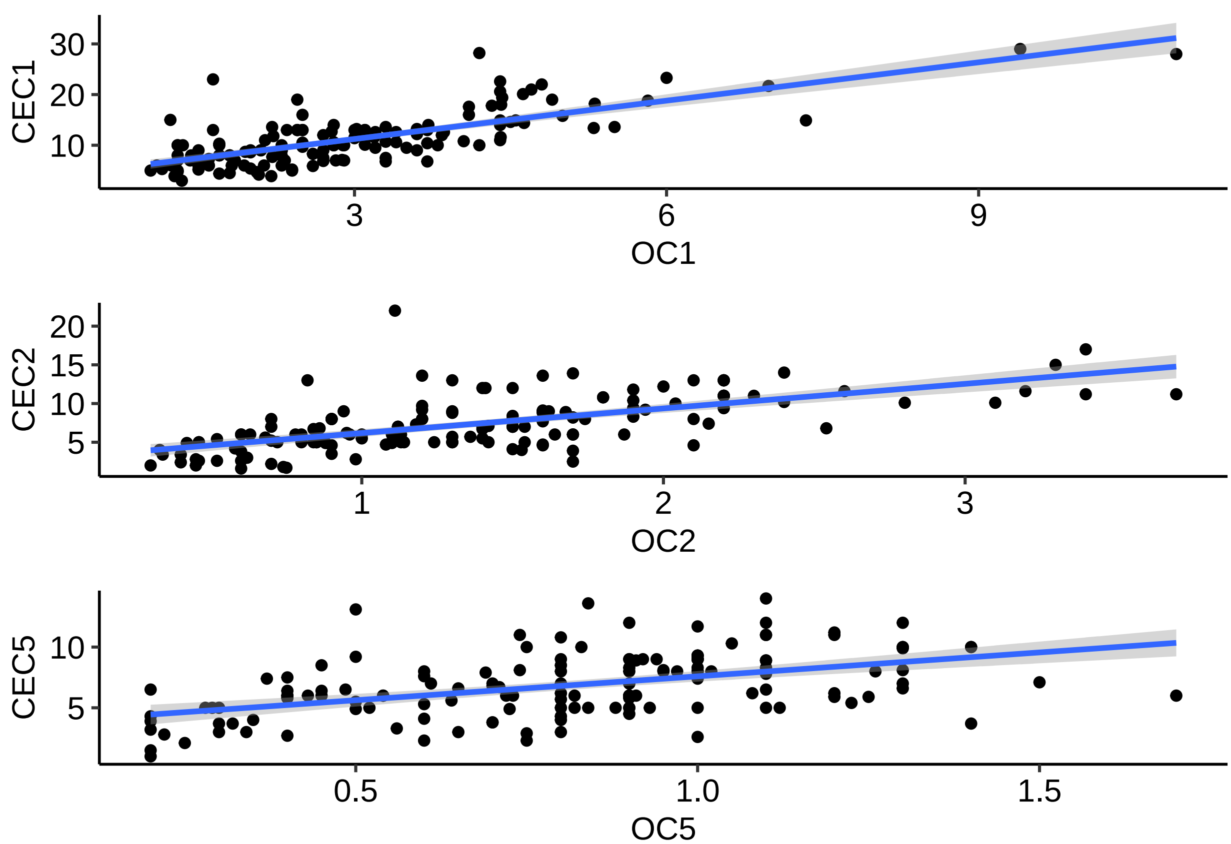
```
## # A tibble: 18 x 2
##     name  p_value
##    <int>    <dbl>
##  1     4 4.65e-27
##  2    11 2.96e-18
##  3    12 8.79e-18
##  4     5 1.17e-15
##  5     7 5.35e-15
##  6     8 6.40e-14
```

```
##   7      1 2.11e-13
##   8      9 9.23e-11
##   9     18 6.58e-10
## 10     14 6.58e- 9
## 11     13 1.46e- 8
## 12      6 5.91e- 8
## 13     15 6.73e- 8
## 14      2 2.63e- 7
## 15      3 2.55e- 6
## 16     17 3.09e- 4
## 17     10 3.61e- 4
## 18     16 2.30e- 1
```

5.2 Remarks

- We get the exact same sequence of models like when we order the models by descending R^2-value
- Why? Specifically for a single explanatory variable (Y = a + bX + e), there is a mathematical relationship between these two values

```
figures$figure4 <- ggarrange(plots$plot4, plots$plot11, plots$plot18,
                             ncol = 1, nrow = 3)
figures$figure4
```



5.3 Remarks

- By looking at the best three plots we can see, that there is a small confidence interval

6 Based on the R-squared value, what is the best predictor for the sub-soil CEC value, given top-soil samples of Clay, OC and CEC? Did you expect this outcome? What is the straight-line equation of this predictor?

CEC5 ~ Clay1: Model 13

```
simple_linear_models$sum13$r.squared
```

```
## [1] 0.1992178
```

CEC5 ~ OC1: Model 16

```
simple_linear_models$sum16$r.squared
```

```
## [1] 0.009920686
```

CEC5 ~ CEC1: New Model (model 19)

```
simple_linear_models$lm19 <- lm(CEC5 ~ CEC1, data = soil_tibble)
simple_linear_models$sum19 <- summary(simple_linear_models$lm19);
simple_linear_models$sum19$r.squared
```

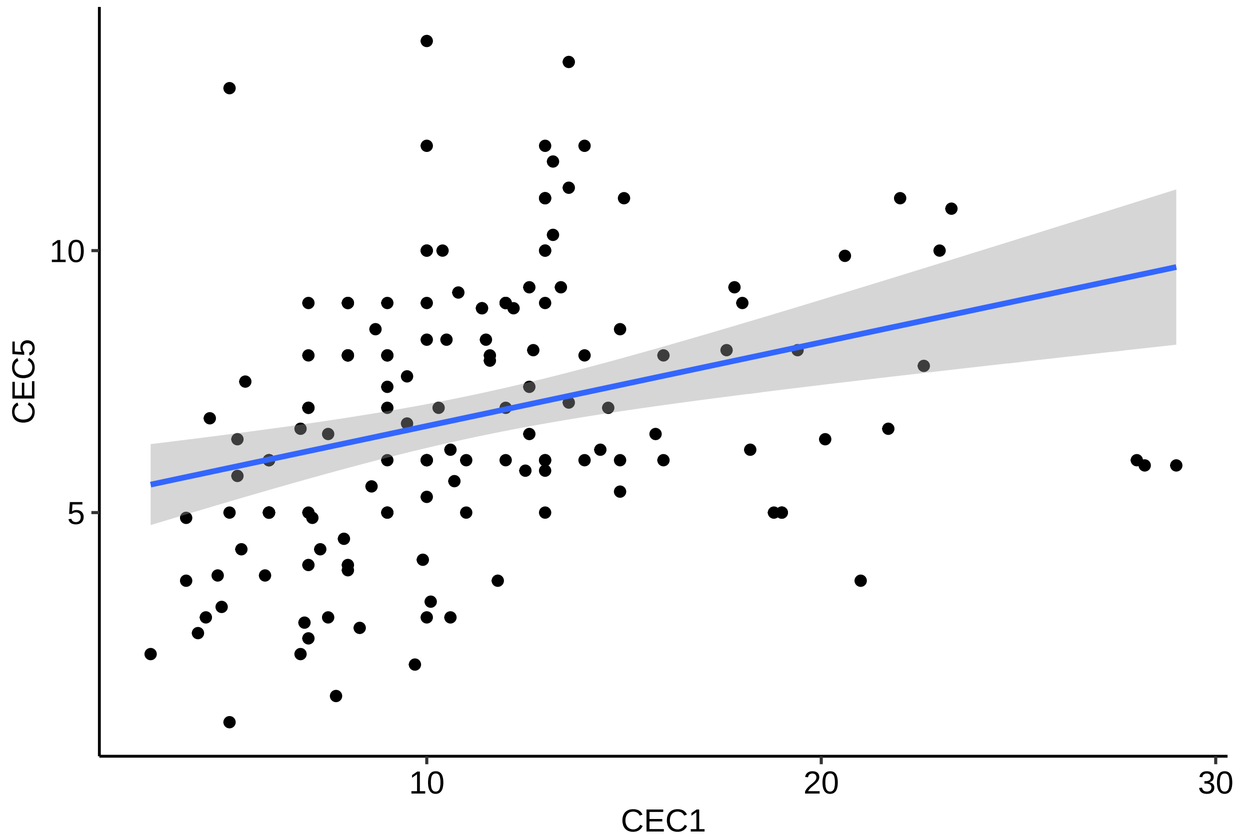
```
## [1] 0.09678499
```

6.1 Remarks

- When we compare these three R^2 values, we can observe that Clay1 has the biggest effect on CEC5 levels
- OC1 has the lowest effect on CEC5 levels
- This could suggest that organic carbon particles rather move upwards than downward
- Or clay rather moves downward compared to organic carbon

Plot model 19

```
plots$plot19 <- ggplot(data=soil_tibble, aes(x=soil_tibble$CEC1, y=soil_tibble$CEC5)) +
  geom_point() + geom_smooth(method = 'lm') + xlab('CEC1') + ylab('CEC5')
plots$plot19
```



Put Clay1, OC1 and CEC1 together in one model to predict CEC5 (model20)

```
simple_linear_models$lm20 <- lm(CEC5 ~ Clay1 + CEC1 + OC1, data = soil_tibble)
simple_linear_models$sum20 <- summary(simple_linear_models$lm20);
simple_linear_models$sum20$r.squared
```

```
## [1] 0.3091561
```

6.2 Remarks

- Model 20 (CEC5 ~ Clay1, CEC1, OC1) has an R^2 value of 0.309
- Why? Because if the top layers in a soil contain a high amount of positive ions, usually lower layers contain high amounts too
- Additionally in soils ions can move downward ("washed out"). But they can also move toward lower levels. Or even upward

Straight-line equation for predictor CEC5 = a + b * Clay1 + c * CEC1 + d * OC1

```
simple_linear_models$sum20
```

```
##
## Call:
## lm(formula = CEC5 ~ Clay1 + CEC1 + OC1, data = soil_tibble)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.7688 -1.3957 -0.1917  1.4533  6.5020
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.11729    0.49268   8.357 5.14e-14 ***
## Clay1        0.09982    0.01665   5.997 1.56e-08 ***
## CEC1         0.20223    0.05454   3.708 0.000298 ***
## OC1         -0.89063    0.19164  -4.647 7.56e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.196 on 143 degrees of freedom
## Multiple R-squared:  0.3092, Adjusted R-squared:  0.2947
## F-statistic: 21.33 on 3 and 143 DF,  p-value: 1.775e-11
```

CEC5 = 4.12 + 0.10 * Clay1 + 0.20 * CEC1 - 0.89 * OC1

7 Do a residual plot for this predictor and interpret it.

Save predicted and residual values

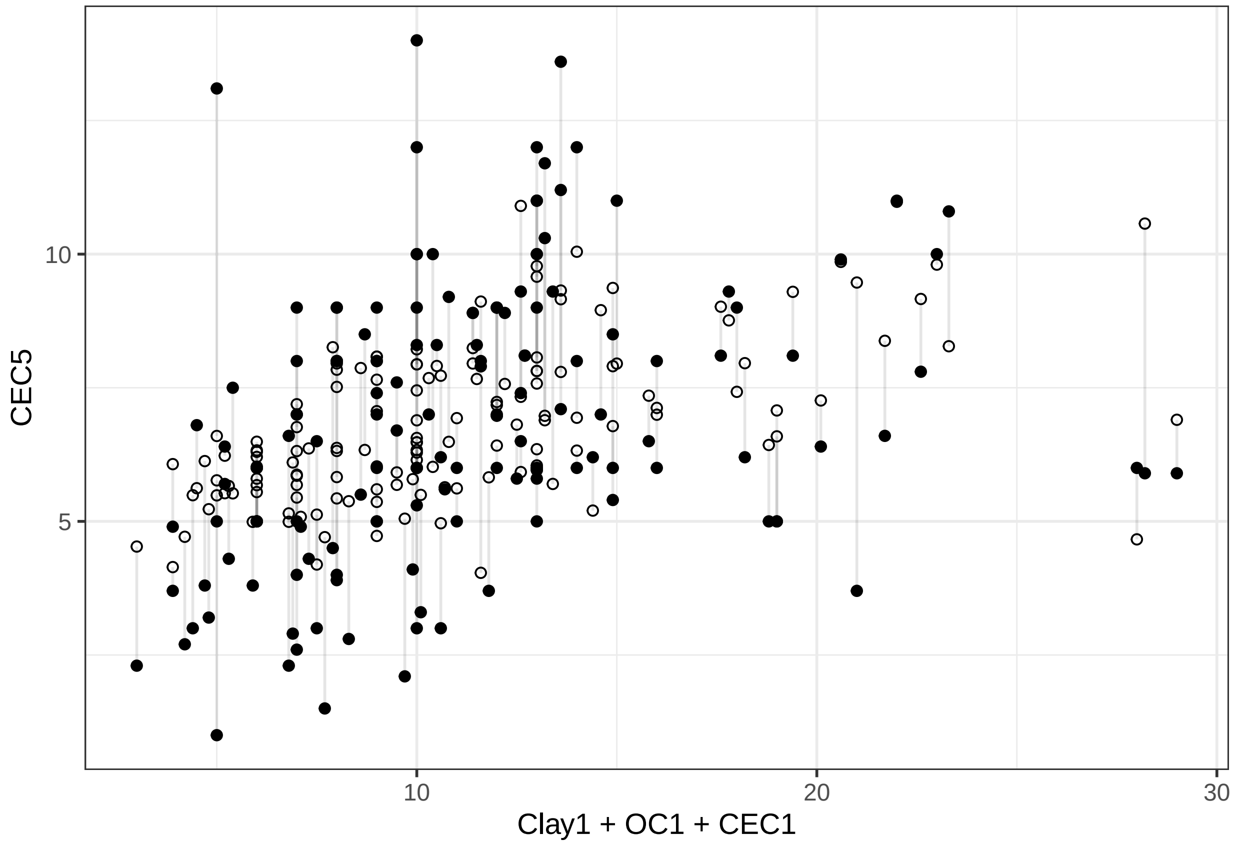
```r
residuals <- list(
  predict(simple_linear_models$lm20),
  residuals(simple_linear_models$lm20)
)
```

Convert to tibble and rename

```r
residuals <- as.data.frame(residuals)
residuals_tibble <- as_tibble(residuals)
names(residuals_tibble)[1] <- 'predicted'
names(residuals_tibble)[2] <- 'residuals'
```

Create residuals vs. fitted plot

```r
plots$residual_plot <- ggplot(data=soil_tibble, aes(x=soil_tibble$CEC1, y=soil_tibble$CEC5)) +
  geom_point() +
  geom_point(data=residuals_tibble, aes(y = residuals_tibble$predicted), shape = 1) +
  geom_segment(aes(xend = soil_tibble$CEC1, yend = residuals_tibble$predicted), alpha = .1) +
  xlab('Clay1 + OC1 + CEC1') + ylab('CEC5') +
  theme_bw()
plots$residual_plot
```



7.1 Remarks

- There is a slightly non-linear relationship between the residuals and the fitted values
- Maybe it would be better to create a model which uses quadratic predictors too

8 Using the above predictor, what is the sub-soil CEC value predicted for a soil with no topsoil clay? What is the sub-soil CEC value predicted for soil with 70 weight-% of topsoil clay? Plot and interpret your result.

Create list with coefficient values

```
predict <- list(Intercept = simple_linear_models$sum13$coefficients[1],
                Clay1 = simple_linear_models$sum13$coefficients[2])
```

Predict CEC5 with model 13 when there is no top soil clay (Clay1 = 0)

```
predict$Intercept + 0 * predict$Clay1
```

```
## [1] 4.22563
```

Predict CEC5 with model 13 when there is 70% top soil clay (Clay1 = 70.0)

```
predict$Intercept + 70.0 * predict$Clay1
```

```
## [1] 10.08712
```

Make a plot to visualize predictions with confidence interval

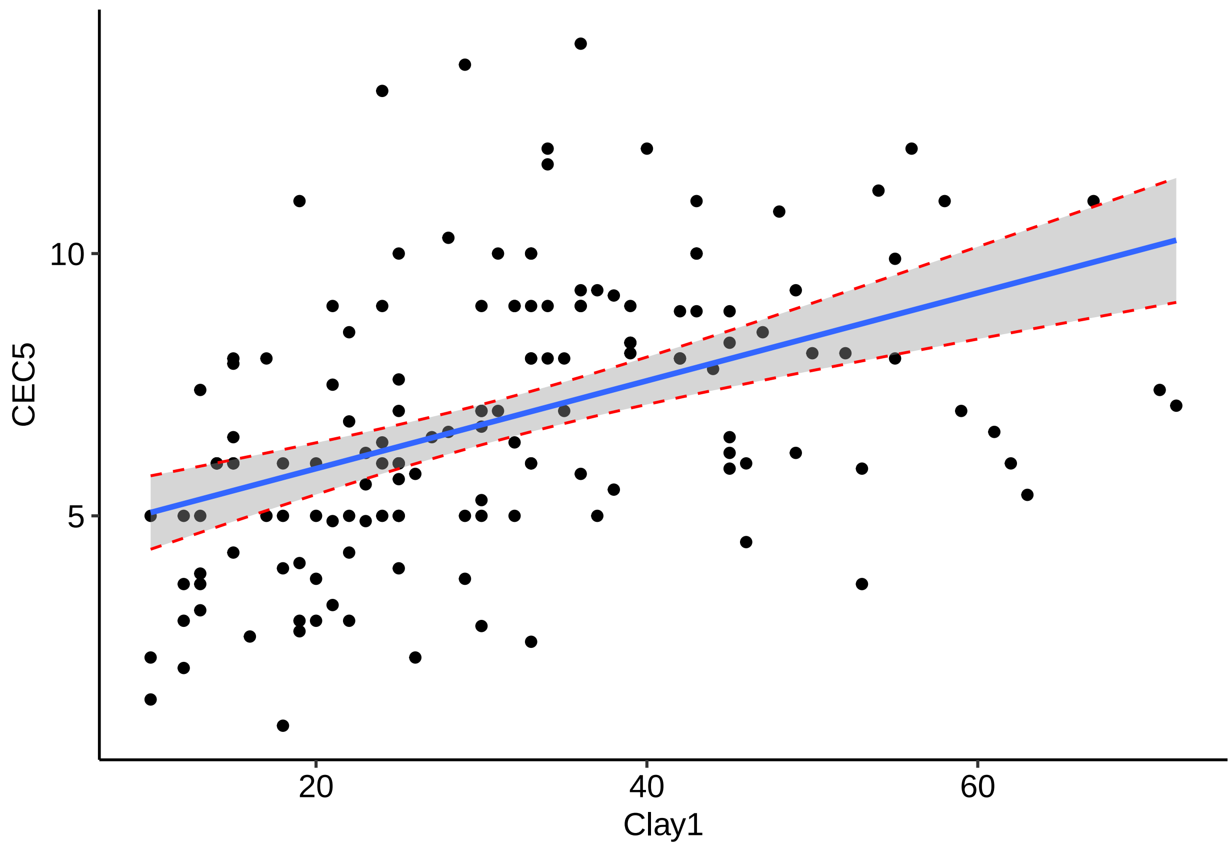
```
predict$predict <- predict(simple_linear_models$lm13, soil_tibble, interval = 'confidence')
predict$data <- cbind(soil_tibble, predict$predict)
```

Regression line + confidence intervals

```
plots$prediction <- ggplot(data=predict$data, aes(x=soil_tibble$Clay1, y=soil_tibble$CEC5)) +
  geom_point() + geom_smooth(method = 'lm') + xlab('Clay1') + ylab('CEC5')
```

Add prediction intervals

```
plots$prediction + geom_line(aes(y = lwr), color = "red", linetype = "dashed") +
  geom_line(aes(y = upr), color = "red", linetype = "dashed")
```



8.1 Remarks

- So we can see by our eyes that for a Clay1 value of 0 there is about 5 of CEC5
- And for an input value of 70 of Clay1 there is about 10 of CEC5

9 What other business-relevant insight could you possibly get from that data set? Try out something, and interpret the results (even if it does not work out!)

One question to answer is if organic carbon is rather moving upward or rather moving downward? to prove this question we create two different models:

Model21: OC5 ~ OC1 + OC2

```
simple_linear_models$lm21 <- lm(OC5 ~ OC1 + OC2, data = soil_tibble)
simple_linear_models$sum21 <- summary(simple_linear_models$lm21)
```

Model22: OC1 ~ OC2 + OC5

```
simple_linear_models$lm22 <- lm(OC1 ~ OC2 + OC5, data = soil_tibble)
simple_linear_models$sum22 <- summary(simple_linear_models$lm22)
```

Examine models 21 and 22

```
simple_linear_models$sum21
```

```
##
## Call:
## lm(formula = OC5 ~ OC1 + OC2, data = soil_tibble)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.77060 -0.12709  0.01481  0.12313  0.50040
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.321945   0.040297   7.989 3.97e-13 ***
## OC1         -0.009459   0.014002  -0.676      0.5
## OC2          0.372868   0.030233  12.333  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2003 on 144 degrees of freedom
## Multiple R-squared:  0.6136, Adjusted R-squared:  0.6082
## F-statistic: 114.3 on 2 and 144 DF,  p-value: < 2.2e-16
```

```
simple_linear_models$sum22
```

```
##
## Call:
## lm(formula = OC1 ~ OC2 + OC5, data = soil_tibble)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.0141 -0.7181 -0.2130  0.4581  7.7770
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.2529     0.2680   4.674 6.71e-06 ***
## OC2           1.4471     0.2276   6.357 2.55e-09 ***
## OC5          -0.3340     0.4944  -0.676      0.5
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.19 on 144 degrees of freedom
## Multiple R-squared:  0.3796, Adjusted R-squared:  0.371
## F-statistic: 44.05 on 2 and 144 DF,  p-value: 1.185e-15
```

9.1 Remarks

- We can see that model 21 has a much higher R^2 value (and also a slightly lower p-value)
- We would argue that organic carbon is rather moving downward than upward
- As a business-relevant insight we would argue that keeping OC1 level high not only leads to high levels of CEC1 but ultimately also to higher levels of OC2 and OC5 (and higher levels of CEC2 and CEC5 thereof)

```r
library("scatterplot3d")
scatterplot3d(soil_tibble$OC1, soil_tibble$OC2, soil_tibble$OC5, highlight.3d=TRUE, col.axis="blue",
              col.grid="lightblue", main='3D Scatterplot', pch=1, angle = 30,
              xlab = 'OC1',
              ylab = 'OC2',
              zlab = 'OC5')
```