Exploratory data analysis

1. Data cleaning

```
library(nlme)
library(readr)
library(vcd)
library(reshape2)
library(lme4)
# load data
Collection <- read_csv("ABMI_Soil_Substrate_Data_2007-2017/A_T23A_Soil_Collection_8696278229856835266.c
Disturbance <- read_csv("ABMI_Soil_Substrate_Data_2007-2017/A_T23B_Soil_Disturbance_6115164298572665863
Mineral <- read_csv("ABMI_Soil_Substrate_Data_2007-2017/A_T25_Mineral_Soil_408583918077381328.csv")
# transfering string data type into factors
Collection <- as.data.frame(unclass(Collection),check.names = FALSE)
Disturbance <- as.data.frame(unclass(Disturbance),check.names = FALSE)
Mineral <- as.data.frame(unclass(Mineral),check.names = FALSE)
# remove missing data
Collection <- Collection[!is.na(Collection$`ABMI Site`),]
Disturbance <- Disturbance[!is.na(Disturbance$`ABMI Site`),]
Mineral <- Mineral[!is.na(Mineral$`ABMI Site`),]
Mineral <- Mineral[!Mineral$`Total Carbon (Percent of Dry Weight)`=='VNA',]
Mineral <- Mineral[Mineral$Quadrant!='DNC',]
Mineral = droplevels(Mineral)
#
#Test whether the carbon in four direction are same
# rotation 1
Mineral1 = Mineral[Mineral$Rotation=='Rotation 1',]
# rotation 2
Mineral2 = Mineral[Mineral$Rotation=='Rotation 2',]
M1 = matrix(NA,length(unique(Mineral1$`ABMI Site`)),4)
colnames(M1) = levels(Mineral1$Quadrant)
rownames(M1) = sort(unique(Mineral1$`ABMI Site`))
M2 = matrix(NA,length(unique(Mineral2$`ABMI Site`)),4)
colnames(M2)<-levels(Mineral2$Quadrant)
for(i in rownames(M1)){
  for (j in levels(Mineral$Quadrant)) {
    Carbon = Mineral1[which(Mineral1$`ABMI Site`==i & Mineral1$Quadrant==j),]$`Total Carbon (Percent of
    if(length(Carbon)!=0){
      M1[i,j] = mean(Carbon)
    }
  }
}
### First test whether there is a difference within the four quadrant.
# M1.new=as.matrix(na.omit(M1))
# max=apply(M1.new, 1, max)
# min=apply(M1.new, 1, min)
# # diff = max - min
# M1.new=cbind(M1.new,diff=max-min)
# # using a t-test to test whether the four direction are different.
# t.test(M1.new[,'diff'])

#1. Merge three tables for round1
```

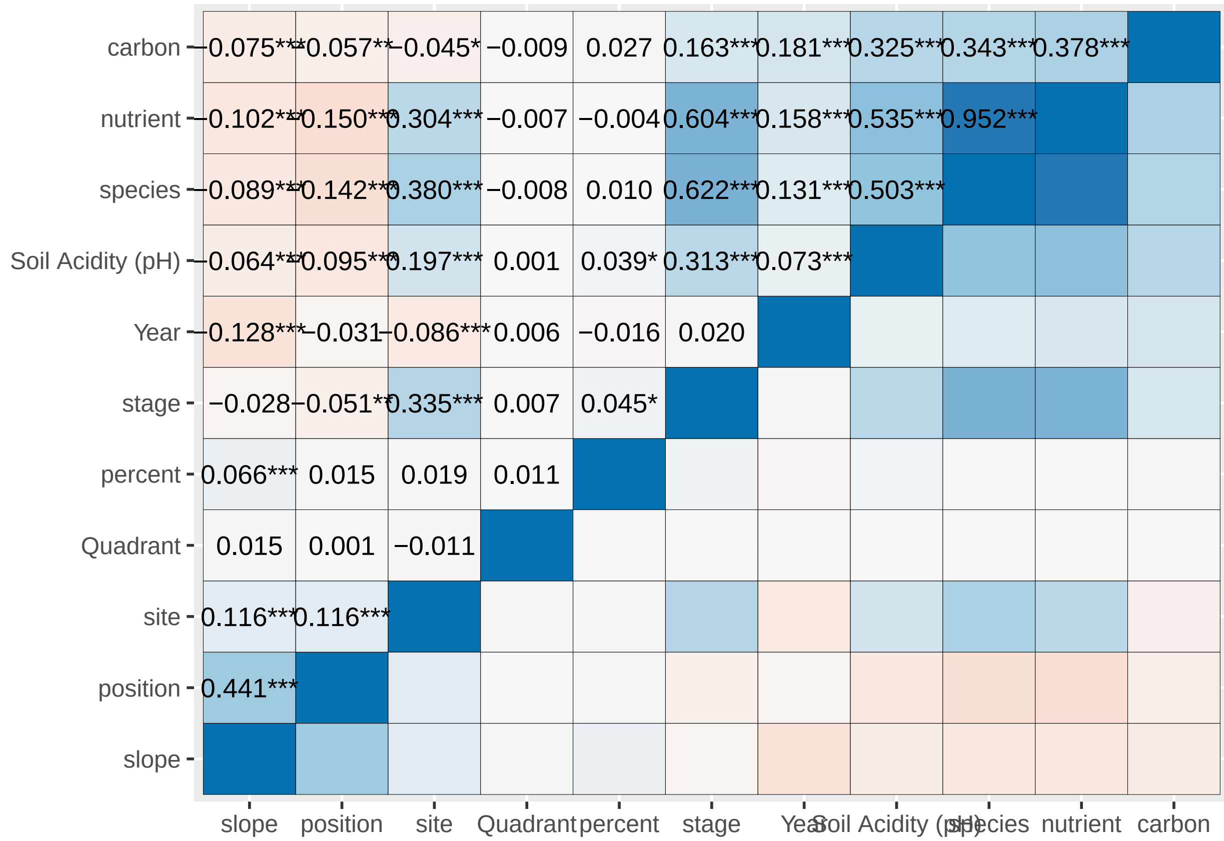
```r
Collection1 = Collection[Collection$Rotation=='Rotation 1',]
#identify columns contain soil core sample
n1=grep("Soil Core Sample", colnames(Collection1))
#romove those columns
Collection1_uni=unique(Collection1[ ,-c(n1)])
# spread mineral
Disturbance1 = Disturbance[Disturbance$Rotation=='Rotation 1',]
Disturbance1.human = Disturbance1[Disturbance1$`Human or Natural Disturbance`=='Human',]
names(Disturbance1.human)[8]<-'Human'
Disturbance1.human$`Human or Natural Disturbance` <- NULL
# merge the collection unique table and mineral table
A=merge(x = Collection1_uni[,-c(4:5)], y = Mineral1[,-c(4:7)], by = c("Rotation","ABMI Site","Year","Qua
# remove missing data
A=A[!is.na(A$`Total Carbon (Percent of Dry Weight)`),]
A$`Total Carbon (Percent of Dry Weight)` = as.numeric(A$`Total Carbon (Percent of Dry Weight)`)
A = A[A$`Slope Position`!='DNC',]
# table 2 cannot be merged because there are multiple levels of disturbance type corresponding to one q
All = A[,c(2:10,15:16)]
All = droplevels(All)
# replace colume name to enable the code more clear.
colnames(All)[1] <- 'site'
colnames(All)[4] <- 'nutrient'
colnames(All)[5] <- 'species'
colnames(All)[6] <- 'stage'
colnames(All)[7] <- 'percent'
colnames(All)[8] <- 'position'
colnames(All)[9] <- 'direction'
colnames(All)[10] <- 'carbon'
# change the slope direction into appropriate category
All$slope = NA
All$direction[All$direction=='VNA'] <- NA
All$direction <- as.numeric(All$direction)
All$slope[which(All$direction < 22.5 | All$direction > 337.5)] <- 'N'
All$slope[which(All$direction > 22.5 & All$direction < 67.5)] <- 'NE'
All$slope[which(All$direction > 67.5 & All$direction < 112.5)] <- 'E'
All$slope[which(All$direction > 112.5 & All$direction < 157.5)] <- 'SE'
All$slope[which(All$direction > 157.5 & All$direction < 202.5)] <- 'S'
All$slope[which(All$direction > 202.5 & All$direction < 247.5)] <- 'SW'
All$slope[which(All$direction > 247.5 & All$direction < 292.5)] <- 'W'
All$slope[which(All$direction > 292.5 & All$direction < 337.5)] <- 'NW'
All$slope[is.na(All$direction)] <- 'FLAT'
All$direction <- NULL
All$slope <- as.factor(All$slope)
levels(All$slope)
```

```
## [1] "E"    "FLAT" "N"    "NE"   "S"    "SE"   "SW"   "W"
```

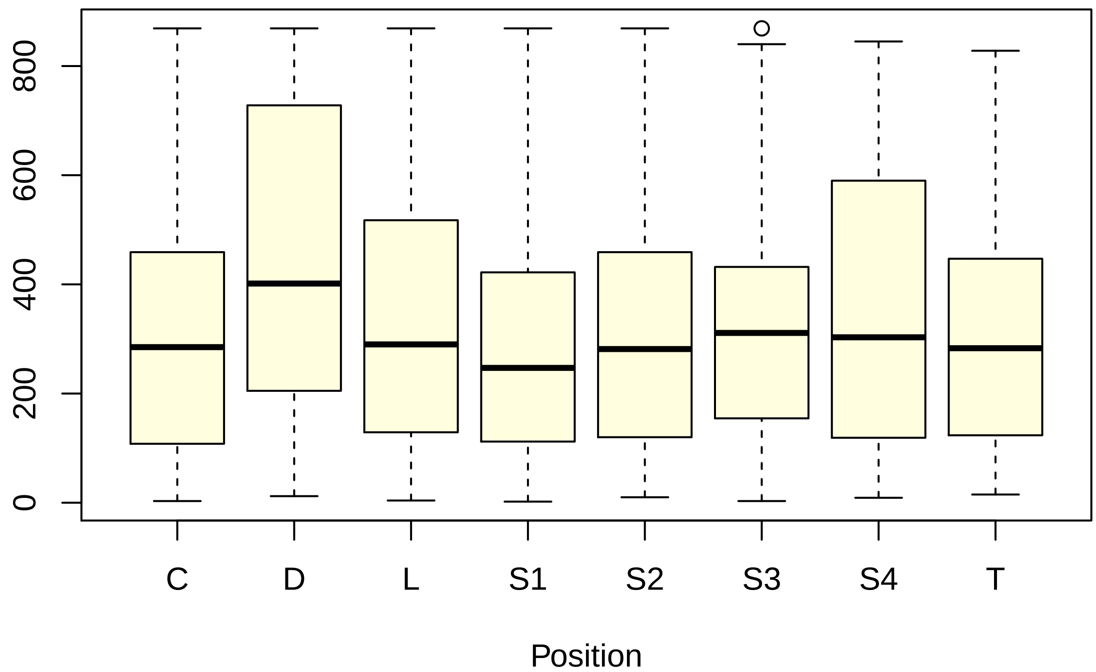
2.Linear mixed effect model

```r
###############################MODEL FITTING################################
library('lattice')
library('kader')
library(MASS)
library(nlme)
library(lme4)
```

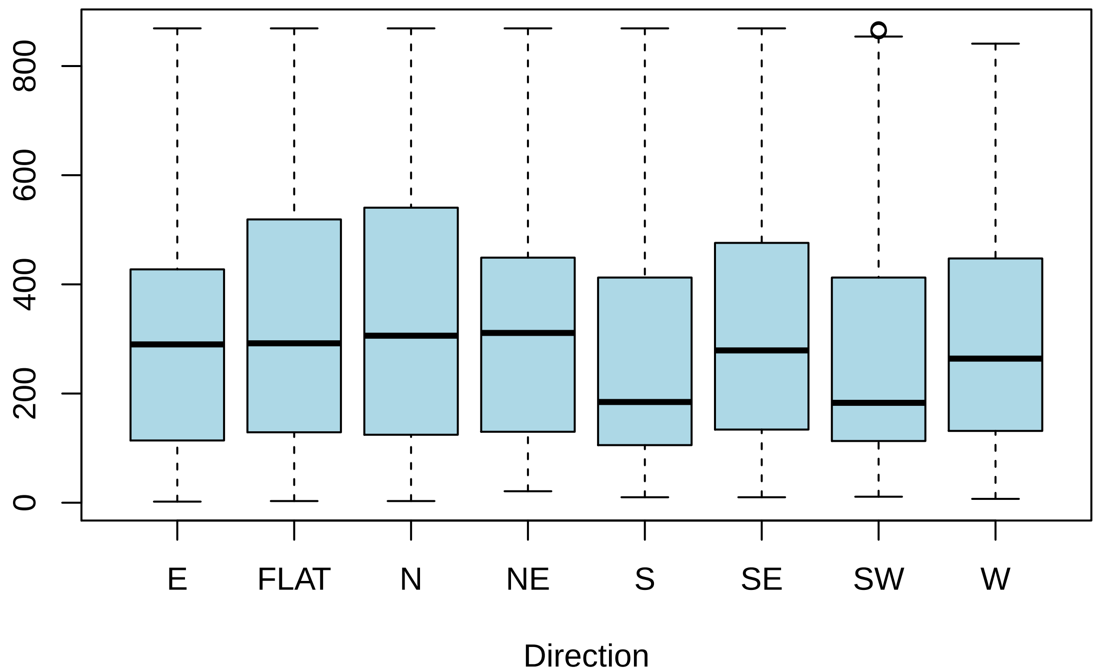
```r
library(sjPlot)
require(car)

# correlation plot to visualize any linear association between carbon and else
DF=All[,c(9,1:8,10,11)]
DF[] <- lapply(DF,as.integer)
sjp.corr(DF)
```



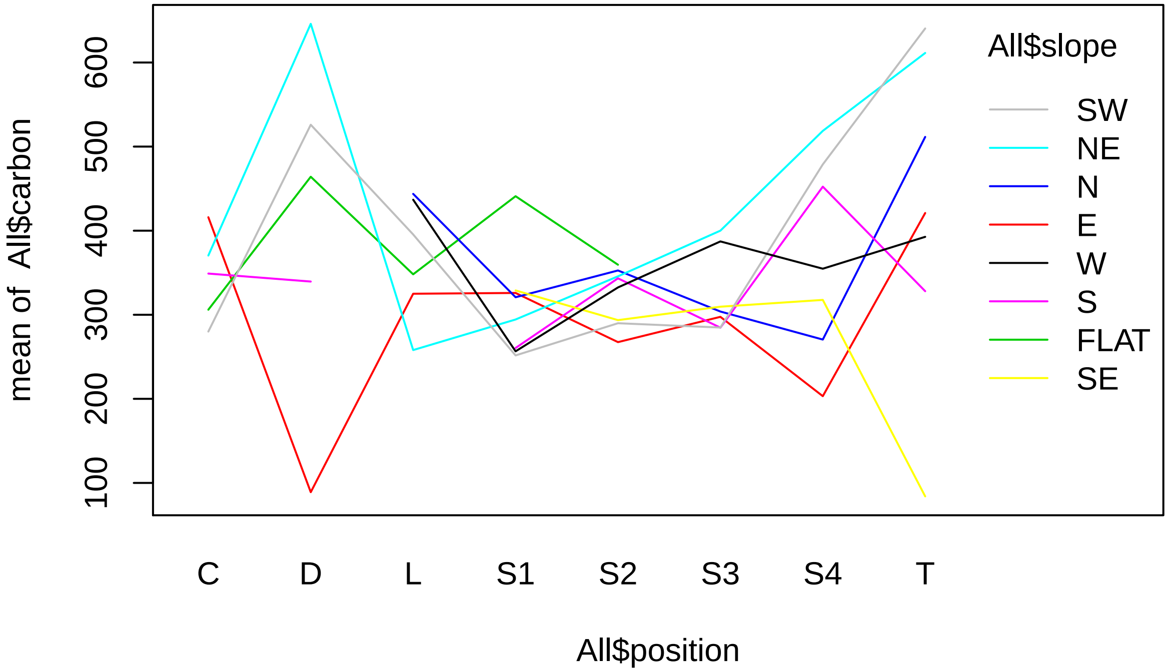

```r
# visualize the data
boxplot(carbon ~ position,col="lightyellow",xlab='Position',All)
```




```
boxplot(carbon ~ slope,col="lightblue",xlab='Direction',All)
```




```
interaction.plot(All$position,All$slope,All$carbon,col = 2:9, lty = 1)
```




```r
#FULL model
lmer1 = lmer(logit(carbon/100) ~ position*slope + (1|site),data = All)
# reduced model 1 (no interaction)
lmer2 = lmer(logit(carbon/100) ~ position+slope+ (1|site),data = All)
# reduced model 2 (no slope direction)
lmer3 = lmer(logit(carbon/100) ~ position + (1|site),data = All)
# reduced model 3 (no position)
lmer4 = lmer(logit(carbon/100) ~ slope + (1|site),data = All)
anova(lmer1,lmer2,lmer3,lmer4)
```

```
## Data: All
## Models:
## lmer3: logit(carbon/100) ~ position + (1 | site)
## lmer4: logit(carbon/100) ~ slope + (1 | site)
## lmer2: logit(carbon/100) ~ position + slope + (1 | site)
## lmer1: logit(carbon/100) ~ position * slope + (1 | site)
##       Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## lmer3 10 6765.8 6826.0 -3372.9   6745.8
## lmer4 10 6804.8 6865.0 -3392.4   6784.8  0.000      0     1.0000
## lmer2 17 6775.9 6878.2 -3371.0   6741.9 42.871      7  3.532e-07 ***
## lmer1 59 6809.2 7164.1 -3345.6   6691.2 50.689     42     0.1682
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(lmer1,lmer2)# interaction effect is not significant
```

```
## Data: All
```

```
## Models:
## lmer2: logit(carbon/100) ~ position + slope + (1 | site)
## lmer1: logit(carbon/100) ~ position * slope + (1 | site)
##       Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## lmer2 17 6775.9 6878.2 -3371.0   6741.9
## lmer1 59 6809.2 7164.1 -3345.6   6691.2 50.689     42     0.1682
```

```r
anova(lmer2,lmer3)# slope direction effect is not significant
```

```
## Data: All
## Models:
## lmer3: logit(carbon/100) ~ position + (1 | site)
## lmer2: logit(carbon/100) ~ position + slope + (1 | site)
##       Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## lmer3 10 6765.8 6826.0 -3372.9   6745.8
## lmer2 17 6775.9 6878.2 -3371.0   6741.9 3.9064      7     0.7905
```

```r
anova(lmer2,lmer4)# position effect is very significant
```

```
## Data: All
## Models:
## lmer4: logit(carbon/100) ~ slope + (1 | site)
## lmer2: logit(carbon/100) ~ position + slope + (1 | site)
##       Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## lmer4 10 6804.8 6865.0 -3392.4   6784.8
## lmer2 17 6775.9 6878.2 -3371.0   6741.9 42.871      7  3.532e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(lmer4,lmer3)# same conclusion
```

```
## Data: All
## Models:
## lmer4: logit(carbon/100) ~ slope + (1 | site)
## lmer3: logit(carbon/100) ~ position + (1 | site)
##       Df    AIC  BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## lmer4 10 6804.8 6865 -3392.4   6784.8
## lmer3 10 6765.8 6826 -3372.9   6745.8 38.965      0  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
lm1=lm(logit(carbon/100) ~ position+Year,data = All)
summary(lm1)
```

```
##
## Call:
## lm(formula = logit(carbon/100) ~ position + Year, data = All)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.4644 -0.6069  0.1374  0.7667  1.8601
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.643e+02  1.511e+01 -10.871  < 2e-16 ***
## positionD    4.651e-01  1.490e-01   3.121  0.00182 **
## positionL    1.085e-01  1.150e-01   0.944  0.34530
```

```
## positionS1  -2.021e-02  1.185e-01  -0.171  0.86455
## positionS2   7.494e-02  1.303e-01   0.575  0.56537
## positionS3   7.170e-02  1.364e-01   0.526  0.59909
## positionS4  -3.350e-02  2.151e-01  -0.156  0.87622
## positionT    1.098e-01  1.740e-01   0.631  0.52793
## Year         7.978e-02  7.516e-03  10.616  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9853 on 3016 degrees of freedom
## Multiple R-squared:  0.04923,    Adjusted R-squared:  0.04671
## F-statistic: 19.52 on 8 and 3016 DF,  p-value: < 2.2e-16
```

```
logLik(lm1)
```

```
## 'log Lik.' -4242.938 (df=10)
```

```
lm1=lm(logit(carbon/100) ~ position+Year+site,data = All)
summary(lm1)
```

```
##
## Call:
## lm(formula = logit(carbon/100) ~ position + Year + site, data = All)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.2468 -0.6340  0.1102  0.7537  2.0178
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.704e+02  1.509e+01 -11.292  < 2e-16 ***
## positionD    4.536e-01  1.484e-01   3.057  0.00225 **
## positionL    1.198e-01  1.145e-01   1.046  0.29561
## positionS1  -3.708e-02  1.180e-01  -0.314  0.75330
## positionS2   7.103e-02  1.298e-01   0.547  0.58414
## positionS3   2.902e-02  1.360e-01   0.213  0.83102
## positionS4  -6.755e-02  2.142e-01  -0.315  0.75249
## positionT    9.423e-02  1.732e-01   0.544  0.58649
## Year         8.271e-02  7.502e-03  11.025  < 2e-16 ***
## site         2.015e-04  3.792e-05   5.316 1.14e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9809 on 3015 degrees of freedom
## Multiple R-squared:  0.05806,    Adjusted R-squared:  0.05525
## F-statistic: 20.65 on 9 and 3015 DF,  p-value: < 2.2e-16
```

```
lm1=lm(logit(carbon/100) ~ position+Year+as.factor(site),data = All)

# the variation between site to site should be considered in the model
# adding the effect of Year
lmer2.y = lmer(logit(carbon/100) ~ position + Year + slope + (1|site),data = All)
lmer3.y = lmer(logit(carbon/100) ~ position + Year + (1|site),data = All)
anova(lmer2,lmer2.y)
```

```
## Data: All
```

```
## Models:
## lmer2: logit(carbon/100) ~ position + slope + (1 | site)
## lmer2.y: logit(carbon/100) ~ position + Year + slope + (1 | site)
##         Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## lmer2   17 6775.9 6878.2 -3371.0   6741.9
## lmer2.y 18 6732.8 6841.1 -3348.4   6696.8 45.099      1  1.873e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(lmer3,lmer3.y) # it impoves the model fit significantly
```

```
## Data: All
## Models:
## lmer3: logit(carbon/100) ~ position + (1 | site)
## lmer3.y: logit(carbon/100) ~ position + Year + (1 | site)
##         Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## lmer3   10 6765.8 6826.0 -3372.9   6745.8
## lmer3.y 11 6725.2 6791.4 -3351.6   6703.2 42.636      1  6.594e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```