# Discovering Associations

Zane Kliesmete

## Count data of rose vase life days.

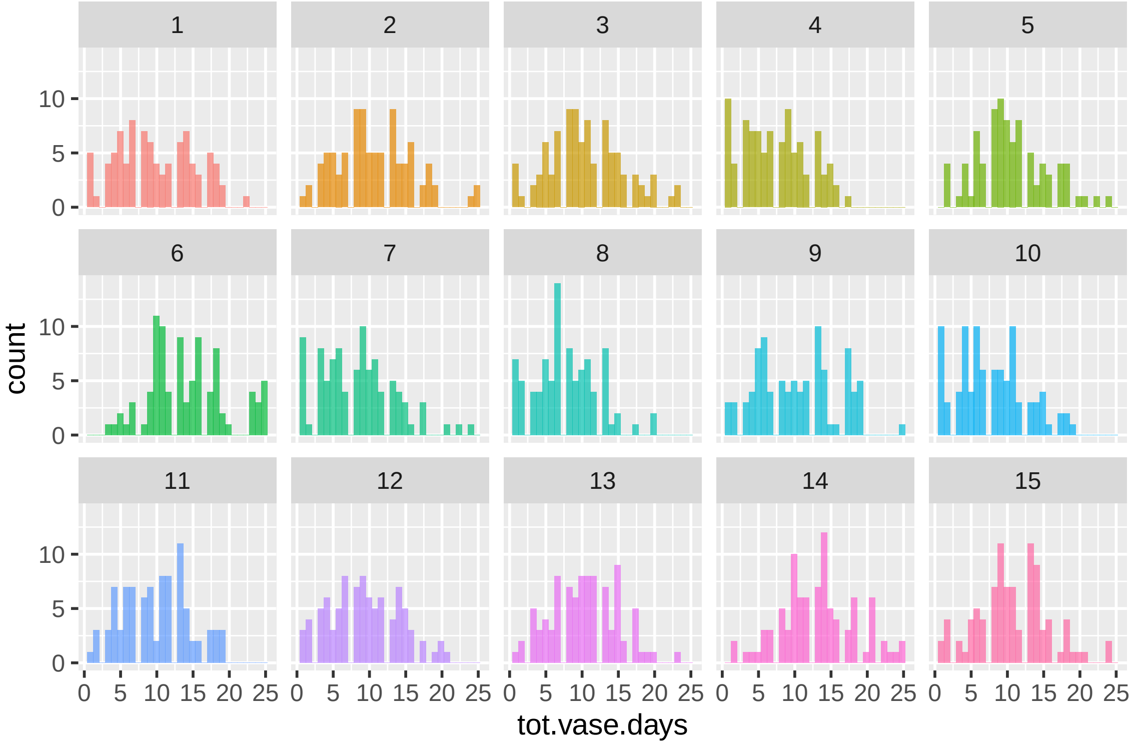
### Poisson-mixed effects modelling.

Out of 1440 datapoints, we have 60 missing outcomes (total vase days), which is 4.1% of the data. More data description, can copy some from simulation description.

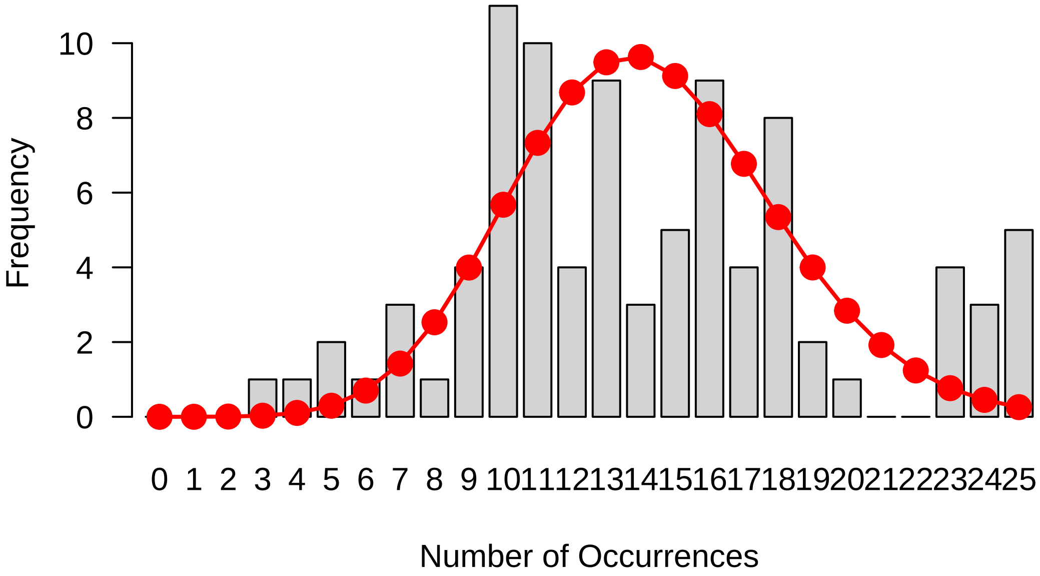
Checking the distribution of the Vase Days of the count data. It looks like there is a larger degree of overdispersion than the poisson model currently accounts for (nope, it's just conditional on the compound: the goodfit should be done per compound).

```
ggplot(d, aes(x=tot.vase.days, fill=compound))+geom_histogram(alpha=0.7)+facet_wrap(~compound, ncol=5)+
```

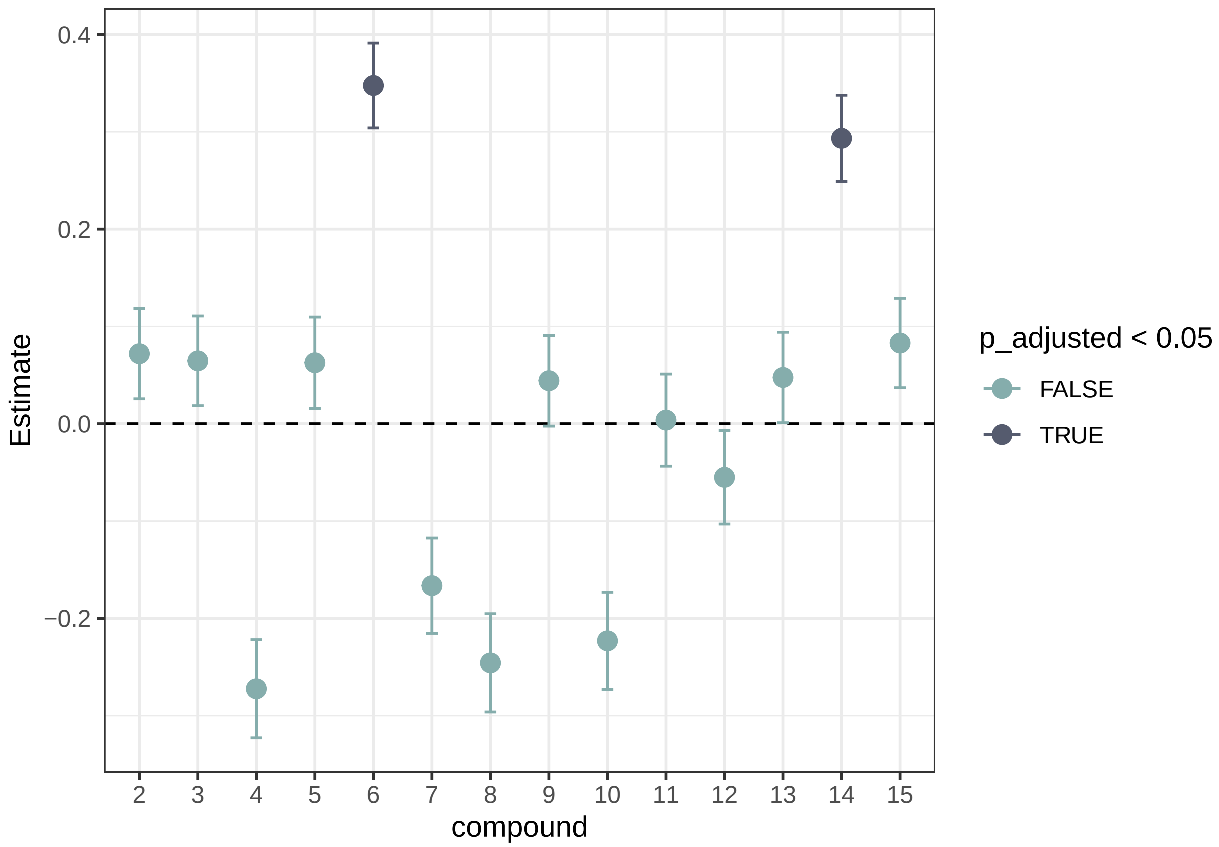
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
#ggplot(d %>% filter(compound %in% c(1, 6, 14)), aes(x=tot.vase.days, fill=as.factor(compound)))+geom_d

# d %>% group_by(compound) %>%
#   dplyr::summarise(mean=mean(tot.vase.days),
#                    var=var(tot.vase.days)) %>%
#   ggplot(aes(x=mean, y=var))+
#   geom_abline(slope=1, intercept=0)+
#   geom_point()


gf <- goodfit(d$tot.vase.days[d$compound==6], "poisson")
plot(gf, type="standing", scale="raw") #I guess the fit would be better if we did it per compound
```



```
#inspired by the first answer here for nested data formulation https://stats.stackexchange.com/question

glmer_out <- glmer(tot.vase.days ~ compound + species + garden + (1|subplotID/bushID), family=poisson(li

glmer_coefficients<-as.data.frame(summary(glmer_out)$coeff) %>%
  rownames_to_column("predictor") %>%
  filter(grepl("compound",predictor)) %>%
  dplyr::rename(pval=`Pr(>|z|)`) %>%
  #we want to have p-adjusted (Holm) values for one-sided test H.alt: lambda(compound)>lambda(water)
  dplyr::mutate(one_sided_pval=ifelse(`z value`>0, pval/2, (1-pval/2)),
         p_adjusted=p.adjust(one_sided_pval, method="holm"),
         significant_higher=ifelse(p_adjusted<0.05, T, F))

ggplot(glmer_coefficients %>%
         mutate(compound=factor(gsub("compound","",predictor), levels=2:15)),
       aes(x=compound, y=Estimate, color=p_adjusted<0.05))+
  geom_hline(yintercept=0, linetype="dashed")+
  geom_errorbar(aes(ymin=Estimate - `Std. Error`, ymax=Estimate +`Std. Error`), width=0.2)+geom_point(s
  scale_color_manual(values=c("#85ADAC","#555B6E"))
```



Conclusion: compounds 6 and 14 significantly increase rose vase days (mention estimates +- sd error, also backcalculated in days, alpha, one-sided Wald test, maybe the exact z and p-values, correction Holm).

### Binomial longitudinal modelling.

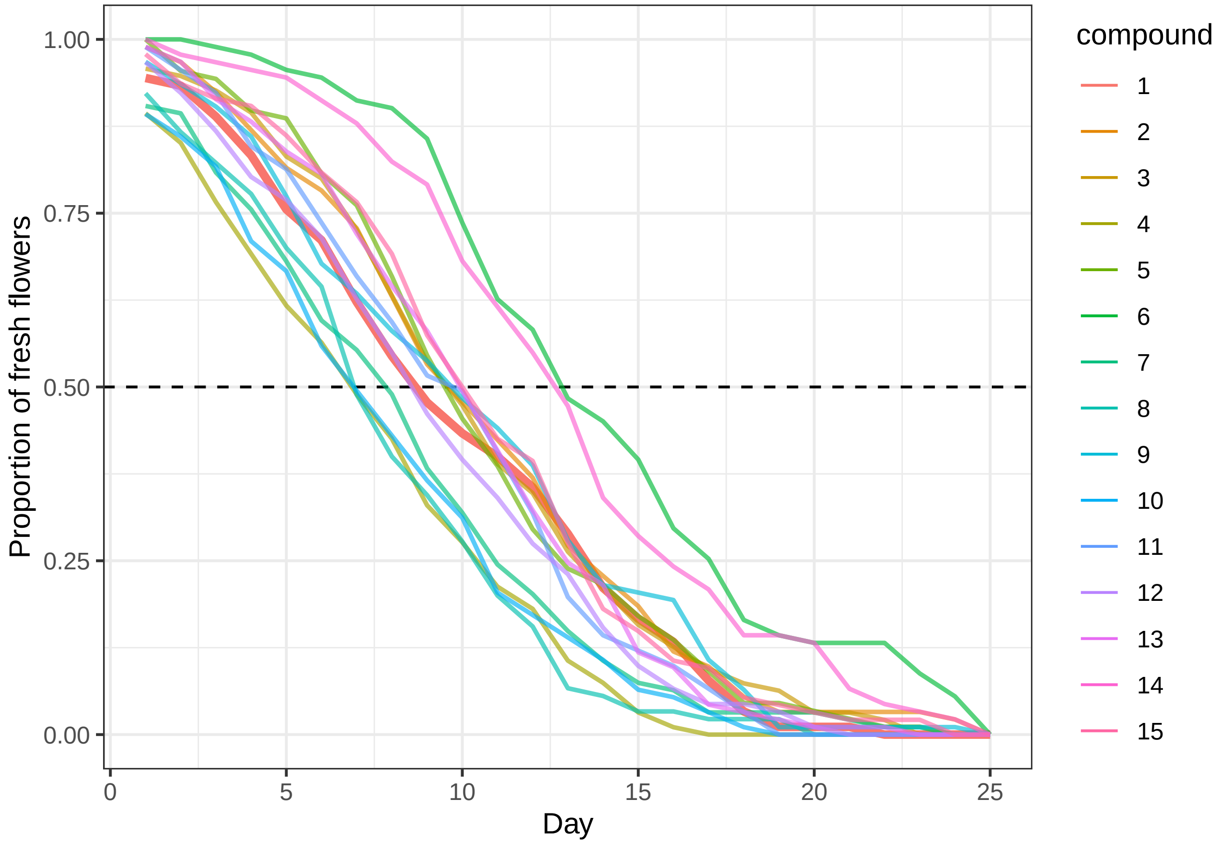
Fit a longitudinal binary data predicting vase life. First need to transform the data into a binary outcome per day.

```
outmat<-matrix(nrow = nrow(d), ncol=max(d$tot.vase.days))

outmat[is.na(outmat)]<-1
for (i in 1:nrow(outmat)){
  outmat[i,c(d[i,tot.vase.days]:25)]<-0
}

outdf<-as.data.frame(outmat)
names(outdf)<-paste0("newVar_",names(outdf))

d_full<-d %>%
  bind_cols(outdf %>% as.data.frame()) %>%
  pivot_longer(contains("newVar"), names_to="day", values_to = "fresh") %>%
  mutate(day=as.numeric(gsub("newVar_V","",day)))
```

```
data_full_cc <- aggregate(fresh ~ compound + day, data = d_full, FUN = mean) %>%
  mutate(water=ifelse(compound==1,T,F))

ggplot(data = data_full_cc)+
  geom_hline(yintercept=0.5, linetype="dashed")+
    geom_line(aes(x = day, y = fresh, color = compound, size=water, alpha=water)) +
  scale_size_discrete(range=c(0.8,1.5),guide="none")+
  scale_alpha_discrete(range=c(0.65,1), guide="none")+
  theme_bw()+
  ylab("Proportion of fresh flowers")+
  xlab("Day")
```

```
## Warning: Using size for a discrete variable is not advised.
```

```
## Warning: Using alpha for a discrete variable is not advised.
```



```
#included this just to see if all datapoints are there..
#ggplot(data = data_full_cc)+geom_line(aes(x = day, y = fresh, color = compound))+facet_wrap(~compound)
```

```
#d_full$fresh<-factor(d_full$fresh, levels=c(1,0))
#d_full$fresh<-relevel(d_full$fresh, ref=1)
glmer_out_bn <- glmer(fresh ~ compound + day + compound*day + species + garden + (1|rater) + (1|subplot
## Warning in (function (fn, par, lower = rep.int(-Inf, n), upper = rep.int(Inf, :
## failure to converge in 10000 evaluations
```

```
## Warning in optwrap(optimizer, devfun, start, rho$lower, control = control, :
## convergence code 4 from Nelder_Mead: failure to converge in 10000 evaluations
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.100509 (tol = 0.002, component 1)
```

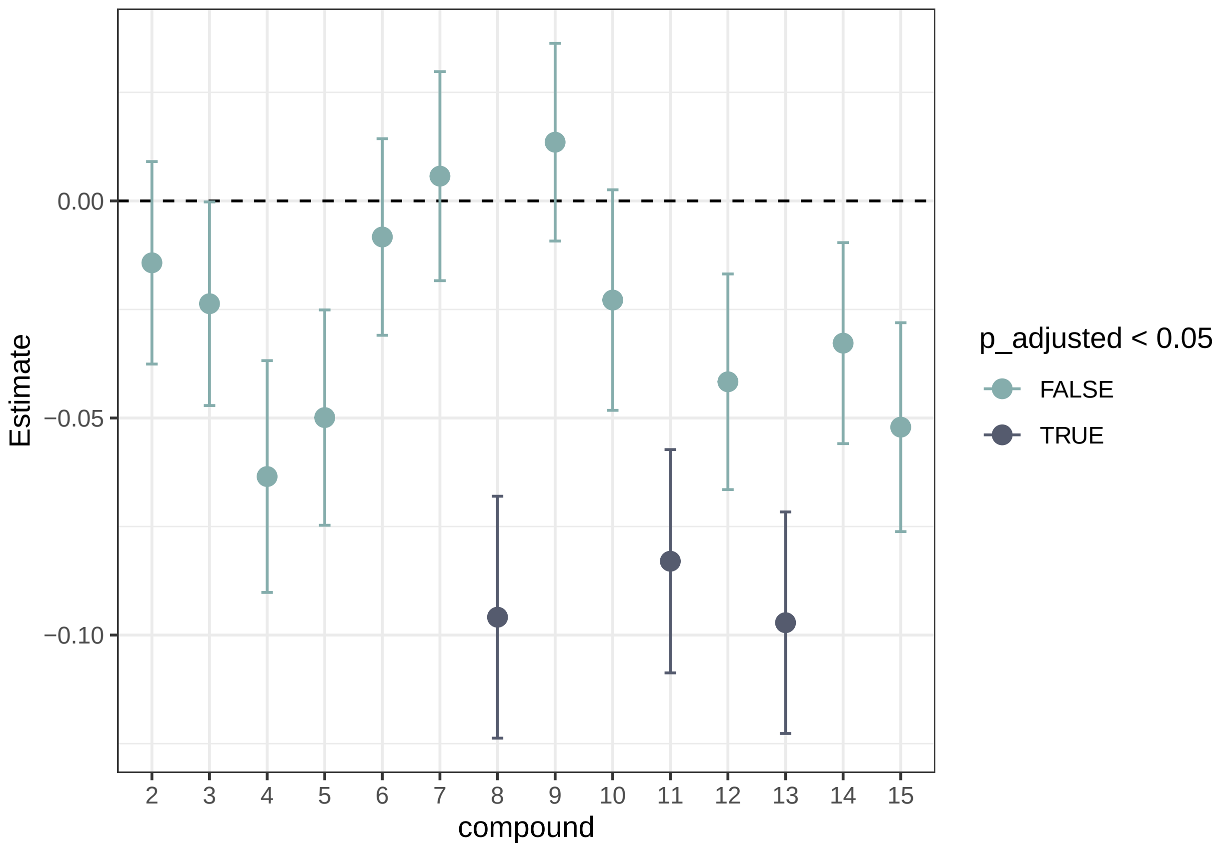
```
summary(glmer_out_bn)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: fresh ~ compound + day + compound * day + species + garden +
##     (1 | rater) + (1 | subplotID/bushID)
##    Data: d_full
##
##      AIC      BIC   logLik deviance df.resid
##  15603.3  15899.0  -7766.7  15533.3    34465
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -16.4789  -0.1729  -0.0246   0.1389  23.5324
##
## Random effects:
##  Groups           Name        Variance Std.Dev.
##  bushID:subplotID (Intercept) 0.2647   0.5145
##  subplotID        (Intercept) 1.4568   1.2070
##  rater            (Intercept) 3.2155   1.7932
## Number of obs: 34500, groups:  bushID:subplotID, 96; subplotID, 16; rater, 6
##
## Fixed effects:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)      4.393157   0.870131   5.049 4.44e-07 ***
## compound2        0.554808   0.263787   2.103 0.035445 *
## compound3        0.608255   0.263288   2.310 0.020876 *
## compound4       -0.849762   0.258916  -3.282 0.001031 **
## compound5        0.864767   0.277123   3.121 0.001805 **
## compound6        2.340682   0.290463   8.058 7.73e-16 ***
## compound7       -0.913807   0.251144  -3.639 0.000274 ***
## compound8       -0.483432   0.270090  -1.790 0.073471 .
## compound9        0.104721   0.255634   0.410 0.682062
## compound10      -0.935279   0.255021  -3.667 0.000245 ***
## compound11       0.840819   0.277611   3.029 0.002456 **
## compound12       0.097540   0.265215   0.368 0.713038
## compound13       1.260931   0.280889   4.489 7.15e-06 ***
## compound14       2.238509   0.287700   7.781 7.21e-15 ***
## compound15       1.022932   0.271378   3.769 0.000164 ***
## day             -0.517386   0.017281 -29.940  < 2e-16 ***
## species2        -0.145855   0.113044  -1.290 0.196964
## garden2          0.936405   0.612957   1.528 0.126591
## compound2:day   -0.014258   0.023319  -0.611 0.540918
## compound3:day  -0.023680   0.023450  -1.010 0.312589
## compound4:day  -0.063486   0.026696  -2.378 0.017400 *
## compound5:day  -0.049909   0.024800  -2.012 0.044171 *
## compound6:day  -0.008311   0.022644  -0.367 0.713590
## compound7:day   0.005702   0.024078   0.237 0.812812
## compound8:day  -0.095884   0.027861  -3.442 0.000578 ***
## compound9:day   0.013524   0.022767   0.594 0.552495
## compound10:day -0.022836   0.025401  -0.899 0.368642
## compound11:day -0.082993   0.025715  -3.227 0.001249 **
## compound12:day -0.041660   0.024837  -1.677 0.093477 .
## compound13:day -0.097151   0.025517  -3.807 0.000141 ***
## compound14:day -0.032761   0.023150  -1.415 0.157024
## compound15:day -0.052113   0.024060  -2.166 0.030316 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Correlation matrix not shown by default, as p = 32 > 12.
## Use print(x, correlation=TRUE)  or
##     vcov(x)        if you need it
```

```
## optimizer (Nelder_Mead) convergence code: 4 (failure to converge in 10000 evaluations)
## Model failed to converge with max|grad| = 0.100509 (tol = 0.002, component 1)
## failure to converge in 10000 evaluations
```

```
glmer_bn_coefficients<-as.data.frame(summary(glmer_out_bn)$coeff) %>%
  rownames_to_column("predictor") %>%
  filter(grepl("compound",predictor)) %>%
  filter(grepl("day",predictor)) %>%
  dplyr::rename(pval=`Pr(>|z|)`) %>%
  #we want to have p-adjusted (Holm) values for one-sided test
  #however in this case it's the moe the day decreases, the more the probability of 1 should increase--
  #H.alt: lambda(compound)<lambda(water)
  dplyr::mutate(one_sided_pval=ifelse(`z value`<0, pval/2, (1-pval/2)),
         p_adjusted=p.adjust(one_sided_pval, method="holm"),
         significant_lower=ifelse(p_adjusted<0.05, T, F))

ggplot(glmer_bn_coefficients %>%
         mutate(compound=factor(gsub("compound|:day","",predictor), levels=2:15)),
       aes(x=compound, y=Estimate, color=p_adjusted<0.05))+
  geom_hline(yintercept=0, linetype="dashed")+
  geom_errorbar(aes(ymin=Estimate - `Std. Error`, ymax=Estimate +`Std. Error`), width=0.2)+geom_point(s
  scale_color_manual(values=c("#85ADAC","#555B6E"))
```



## Gaussian data of flower width.

Gaussian outcome data. We received data from 180 flowers. This was distributed as 12 flowers for each compound for each of the 15 compounds. In each of those groups, there were 6 flowers per species and 6 grown in each garden. There were also 18 different subplots. The number of subplots is greater than the number of number of flowers per group.

For each of the 18 flowers, we have measurements of the width of the flower over the course of 21 days. All measurements for all flowers were taken by a single rater.

Below I transform the data so that there is a row for each measurement of each flower on each day resulting in 3780 rows.

```
g <- fread('gaussian_data_G6.csv')
summary(g) #there is only one rater, drop it
```

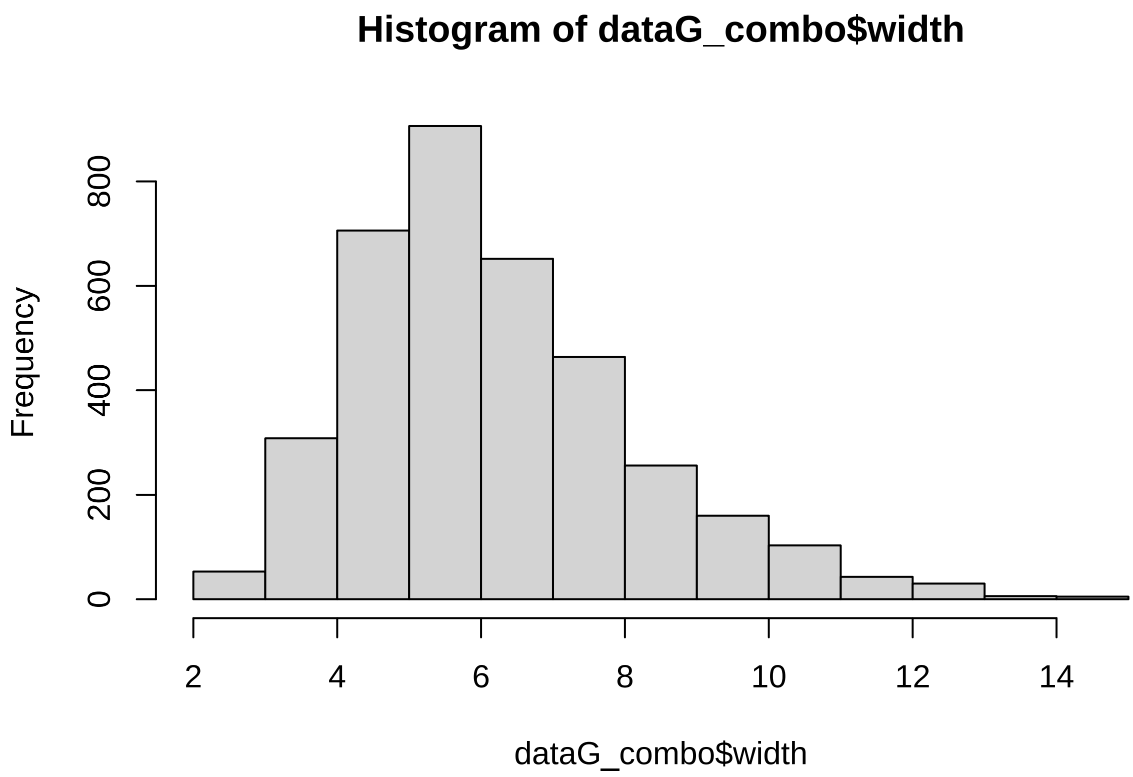
```
##   Flower_index        T_0             T_1             T_2       
##  Min.   :18006   Min.   :2.100   Min.   :2.300   Min.   :2.300  
##  1st Qu.:18206   1st Qu.:3.800   1st Qu.:4.000   1st Qu.:4.100  
##  Median :18498   Median :4.500   Median :4.600   Median :4.700  
##  Mean   :18484   Mean   :4.438   Mean   :4.562   Mean   :4.806  
##  3rd Qu.:18715   3rd Qu.:5.000   3rd Qu.:5.200   3rd Qu.:5.600  
##  Max.   :18988   Max.   :6.700   Max.   :7.500   Max.   :7.000  
##                  NA's   :1       NA's   :1       NA's   :1      
##       T_3             T_4             T_5             T_6       
##  Min.   :2.500   Min.   :2.500   Min.   :3.000   Min.   :3.300
##  1st Qu.:4.300   1st Qu.:4.500   1st Qu.:4.500   1st Qu.:4.950
##  Median :4.900   Median :5.300   Median :5.200   Median :5.500
##  Mean   :4.939   Mean   :5.238   Mean   :5.325   Mean   :5.669
##  3rd Qu.:5.500   3rd Qu.:6.000   3rd Qu.:6.000   3rd Qu.:6.550
##  Max.   :7.300   Max.   :8.100   Max.   :8.300   Max.   :8.800
##  NA's   :1       NA's   :1       NA's   :1       NA's   :1
##       T_7             T_8             T_9              T_10
##  Min.   :2.900   Min.   :2.400   Min.   : 2.900   Min.   : 3.300
##  1st Qu.:4.800   1st Qu.:5.000   1st Qu.: 5.100   1st Qu.: 5.200
##  Median :5.700   Median :5.800   Median : 6.000   Median : 6.200
##  Mean   :5.667   Mean   :5.847   Mean   : 6.114   Mean   : 6.264
##  3rd Qu.:6.400   3rd Qu.:6.700   3rd Qu.: 7.100   3rd Qu.: 7.300
##  Max.   :9.300   Max.   :9.100   Max.   :10.200   Max.   :10.100
##  NA's   :2       NA's   :2       NA's   :3        NA's   :3
##       T_11             T_12             T_13             T_14
##  Min.   : 2.400   Min.   : 2.500   Min.   : 2.500   Min.   : 2.500
##  1st Qu.: 5.400   1st Qu.: 5.400   1st Qu.: 5.500   1st Qu.: 5.900
##  Median : 6.200   Median : 6.400   Median : 6.700   Median : 7.050
##  Mean   : 6.354   Mean   : 6.644   Mean   : 6.795   Mean   : 7.109
##  3rd Qu.: 7.200   3rd Qu.: 7.700   3rd Qu.: 7.800   3rd Qu.: 8.075
##  Max.   :10.100   Max.   :12.500   Max.   :11.400   Max.   :10.900
##  NA's   :3        NA's   :3        NA's   :5        NA's   :6
##       T_15             T_16             T_17             T_18
##  Min.   : 3.000   Min.   : 3.000   Min.   : 2.900   Min.   : 3.600
##  1st Qu.: 5.900   1st Qu.: 6.100   1st Qu.: 6.000   1st Qu.: 6.000
##  Median : 6.800   Median : 7.400   Median : 7.400   Median : 7.500
##  Mean   : 7.216   Mean   : 7.455   Mean   : 7.514   Mean   : 7.726
##  3rd Qu.: 8.700   3rd Qu.: 8.700   3rd Qu.: 8.900   3rd Qu.: 9.375
##  Max.   :12.300   Max.   :12.300   Max.   :12.900   Max.   :12.800
##  NA's   :7        NA's   :7        NA's   :10       NA's   :10
##       T_19             T_20           Compound      Rater        Type
##  Min.   : 3.100   Min.   : 3.000   Min.   : 1   Min.   :1   Min.   :1.0
##  1st Qu.: 6.100   1st Qu.: 6.025   1st Qu.: 4   1st Qu.:1   1st Qu.:1.0
##  Median : 7.800   Median : 7.800   Median : 8   Median :1   Median :1.5
##  Mean   : 8.046   Mean   : 8.011   Mean   : 8   Mean   :1   Mean   :1.5
##  3rd Qu.: 9.600   3rd Qu.: 9.300   3rd Qu.:12   3rd Qu.:1   3rd Qu.:2.0
##  Max.   :14.800   Max.   :14.600   Max.   :15   Max.   :1   Max.   :2.0
##  NA's   :10       NA's   :10
##      Garden       Subplot
##  Min.   :1.0   Min.   : 1.0
##  1st Qu.:1.0   1st Qu.: 5.0
##  Median :1.5   Median : 9.5
##  Mean   :1.5   Mean   : 9.5
##  3rd Qu.:2.0   3rd Qu.:14.0
##  Max.   :2.0   Max.   :18.0
##
```

```
g<-g %>% dplyr::select(-Rater)


colnames(g)<-c("flowerID",0:20,"compound","type","garden","subplot")

dataG_long <- gather(g, days, width, "0":"20", factor_key=TRUE) %>%
      mutate(garden=as.factor(garden),
             type=as.factor(type),
             compound=as.factor(compound),
             subplot=as.factor(subplot),
             days=as.numeric(days),
             flowerID=as.factor(flowerID))

head(dataG_long)
```

```
##   flowerID compound type garden subplot days width
## 1    18075        1    1      1       1    1   2.9
## 2    18767        1    1      1       2    1   2.6
## 3    18028        1    1      1       3    1   5.2
## 4    18326        1    1      2       4    1   6.5
## 5    18017        1    1      2       5    1   4.2
## 6    18718        1    1      2       6    1   5.7
```

I also added a column showing the change in the width of the flower so that we can see the change in width per day. It is worth noting that the width of the flower does not uniformly increase, instead it does fluctuate from day to day, decreasing occasionally. Also, there are quite a few missing measurements, we probably should have accounted for this in our sample size calculation?

```
hist(dataG_combo$width)
```

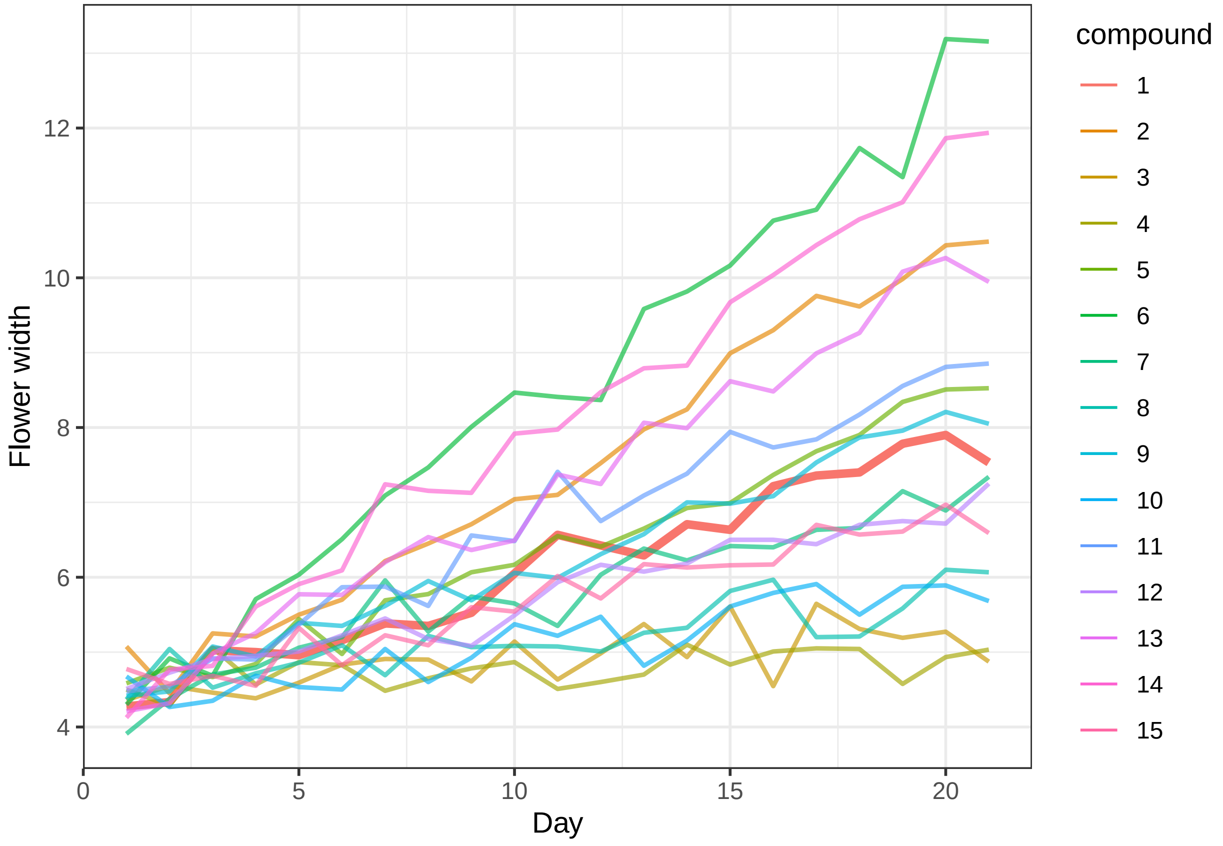


Below I plotted the mean width of the flower by day by compound on a given day.

```
data_cc <- aggregate(width ~ compound + days, data = dataG_combo, FUN = mean) %>%
  mutate(water=ifelse(compound==1,T,F))

ggplot(data = data_cc)+
    geom_line(aes(x = days, y = width, color = compound, size=water, alpha=water)) +
  scale_size_discrete(range=c(0.8,1.5),guide="none")+
  scale_alpha_discrete(range=c(0.65,1), guide="none")+
  theme_bw()+
  ylab("Flower width")+
  xlab("Day")
```

```
## Warning: Using size for a discrete variable is not advised.
```

```
## Warning: Using alpha for a discrete variable is not advised.
```



```
#ggplot(data = data_cc %>% filter(compound %in% c(1, 6, 14)))+
#    geom_line(aes(x = days, y = width, color = compound))
```

The takeaway from this graph is that for each graph, the change in the Width of the flower is not the same for each of the Compounds. Does this mean we have an interaction between Compound and Days?

```
data_ccc<- aggregate(delta_width ~ compound + days, data = dataG_combo, FUN = mean)
```

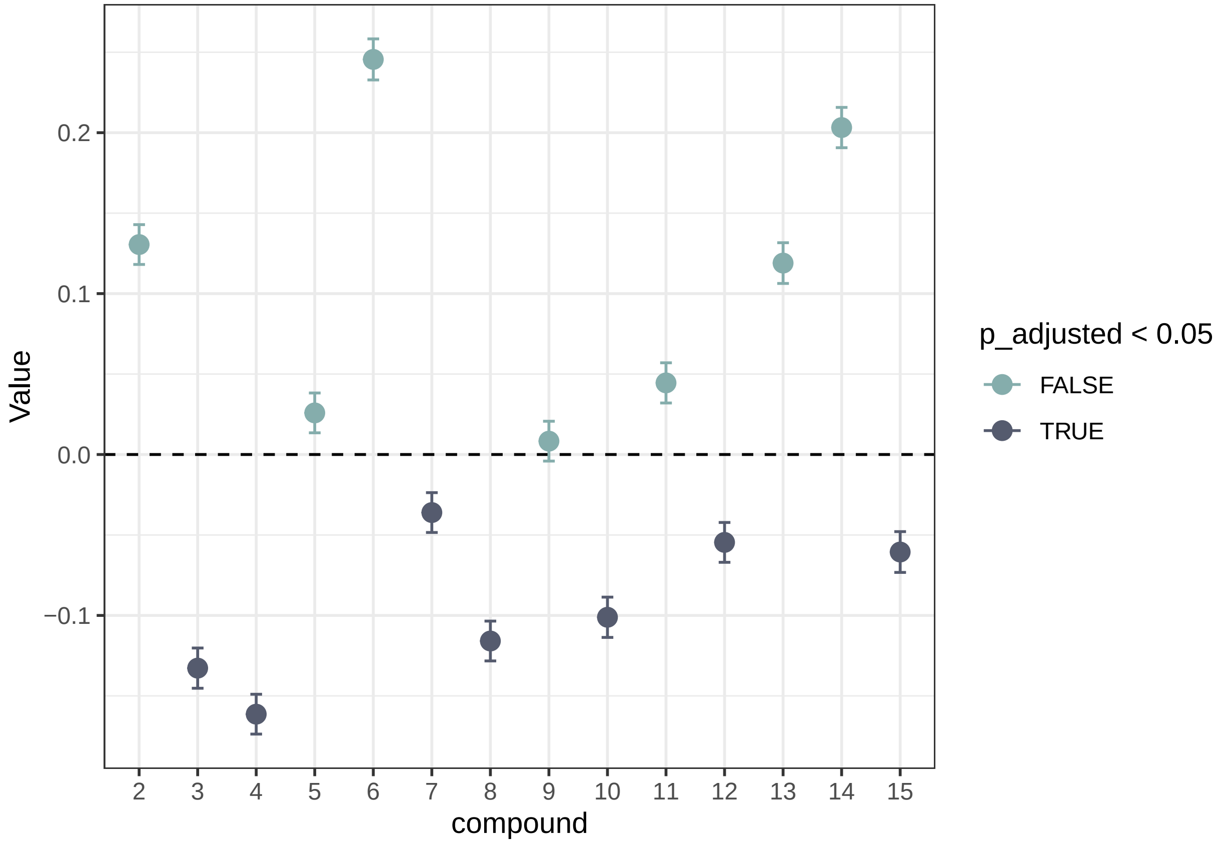
```
plot <- ggplot(data = data_ccc)+
    geom_line(aes(x = days, y = delta_width, color = compound))

plot
```

I fit a linear model to the gaussian outcome data where Compound, Type, Garden and Days are included as fixed effects, a compound and days interaction is included and subplot is included as a random effect. Rater is not included because we only have one rater.

```
#g1 <- glm(Width ~ Compound + Type + Garden + Days + Compound*Days + (1 | Subplot), data=dataG_long)
lme_out <- nlme::lme(width ~ compound + type + garden + days + compound*days, data=dataG_long, random =
```

Probably not right, this output is too long.

```
lme_coefficients<-as.data.frame(summary(lme_out)$tTable) %>%
  rownames_to_column("predictor_full") %>%
  filter(grepl("compound",predictor_full)) %>%
  filter(grepl("days",predictor_full)) %>%
  dplyr::rename(pval=`p-value`) %>%
  #we want to have p-adjusted (Holm) values for one-sided test H.alt: lambda(compound)>lambda(water)
  dplyr::mutate(one_sided_pval=ifelse(`t-value`<0, pval/2, (1-pval/2)),
         p_adjusted=p.adjust(one_sided_pval, method="holm"),
         significant_lower=ifelse(p_adjusted<0.05, T, F),
         predictor=gsub(":days","",predictor_full))

ggplot(lme_coefficients %>%
         mutate(compound=factor(gsub("compound|:day","",predictor), levels=2:15)),
       aes(x=compound, y=Value, color=p_adjusted<0.05))+
  geom_hline(yintercept=0, linetype="dashed")+
  geom_errorbar(aes(ymin=Value - `Std.Error`, ymax=Value +`Std.Error`), width=0.2)+geom_point(size=3)+t
  scale_color_manual(values=c("#85ADAC","#555B6E"))
```

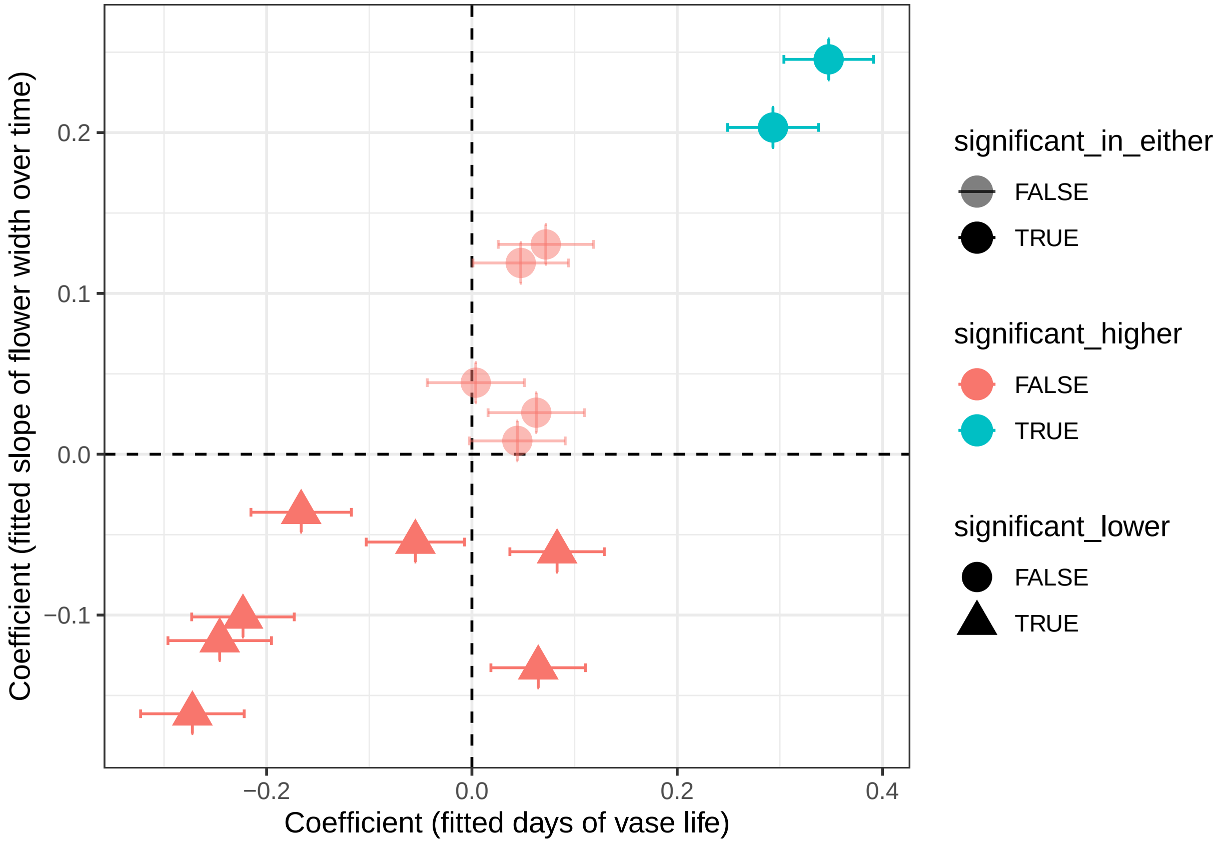


```
# also nice #BEE3DB
```

Now intersect the two model outputs to compare and interpret the results (so far only compared the results of the first and last model).

```
both_predictions<-inner_join(glmer_coefficients, lme_coefficients, by="predictor", suffix=c(".glmer",".l
  mutate(significant_in_either=ifelse(significant_higher | significant_lower, T, F))


ggplot(both_predictions, aes(x=Estimate, y=Value, color=significant_higher, shape=significant_lower, alp
  geom_vline(xintercept = 0, linetype="dashed")+
  geom_hline(yintercept = 0, linetype="dashed")+
  geom_point(size=5)+
  geom_errorbarh(aes(xmin=Estimate-`Std. Error`, xmax=Estimate+`Std. Error`))+
  geom_errorbar(aes(ymin=Value-`Std.Error`, ymax=Value+`Std.Error`))+
  xlab("Coefficient (fitted days of vase life)")+
  ylab("Coefficient (fitted slope of flower width over time)")+
  theme_bw()+
  scale_alpha_discrete(range=c(0.5,1))
```

```
## Warning: Using alpha for a discrete variable is not advised.
```


Coefficient (fitted slope of flower width over time)
0.2
0.1
0.0
−0.1
−0.2
0.0
0.2
0.4
Coefficient (fitted days of vase life)
significant_in_either
FALSE
TRUE
significant_higher
FALSE
TRUE
significant_lower
FALSE
TRUE