env_and_copies_stats

This code runs uni-variate models on parameters of interest.

Getting Started

Libraries

Data

Cleaning

Clean and Trim

```
dust<- dust0 %>%
  filter(!is.na(date),
         date %in% as.Date("2022-07-04"):as.Date("2022-07-19")) %>%
  mutate(t7sum = log10(t7sum)) #Log transform Dust conc (as it is not normal)


data<- data0 %>%
  filter(date %in% as.Date("2022-07-04"):as.Date("2022-07-19"))
```

Combine into one big data set

```
master0<-
  dust %>% full_join(data) %>%
  filter(hr_cst %in% "01") #While this is technically not for hour 1, this gets ride of al
```

```
Joining, by = "date"
```

```
master<- master0[c(1,8,14,34)] #Select columns of interest

#View new dataframe
master
```

```
# A tibble: 40 x 4
   date        t7sum site       copies_mL
   <date>      <dbl> <chr>          <dbl>
 1 2022-07-04 -0.886 <NA>              NA
 2 2022-07-05 -0.796 <NA>              NA
 3 2022-07-06 -0.854 <NA>              NA
 4 2022-07-07 -1     Blind Oso    25462.
 5 2022-07-07 -1     Canals       21658.
 6 2022-07-07 -1     Gulf           442.
 7 2022-07-08 -1     Blind Oso    33586.
 8 2022-07-08 -1     Canals       14290.
 9 2022-07-08 -1     Gulf          2670.
10 2022-07-09 -0.886 Blind Oso    29093.
# ... with 30 more rows
```

Make Site DF

Need to include 7/5 and 7/6 for previous dust data (will be used for lag analysis)

```
bo<- master %>%
  filter(site %in% "Blind Oso"| date %in% c( as.Date("2022-07-05"), as.Date("2022-07-06"))

c2<- master %>%
  filter(site %in% "Canals"| date %in% c( as.Date("2022-07-05"), as.Date("2022-07-06")))

rd<- master %>%
  filter(site %in% "Gulf"| date %in% c(as.Date("2022-07-05"), as.Date("2022-07-06")))
```

DUST

Add Lag from `2_cca.qmd`

```
bo2<- bo %>% mutate(
  lag =Lag(bo$t7sum, shift  = 1)) %>% #Make lag from ccf
filter(between(date, as.Date('2022-07-07'), as.Date('2022-07-19')))

c22<- c2%>% mutate(
  lag =Lag(c2$t7sum, shift  = 2)) %>% #Make lag from ccf
filter(between(date, as.Date('2022-07-07'), as.Date('2022-07-19')))

rd2<- rd%>% mutate(
  lag =Lag(rd$t7sum, shift  = 1)) %>% #Make lag from ccf
filter(between(date, as.Date('2022-07-07'), as.Date('2022-07-19')))
```

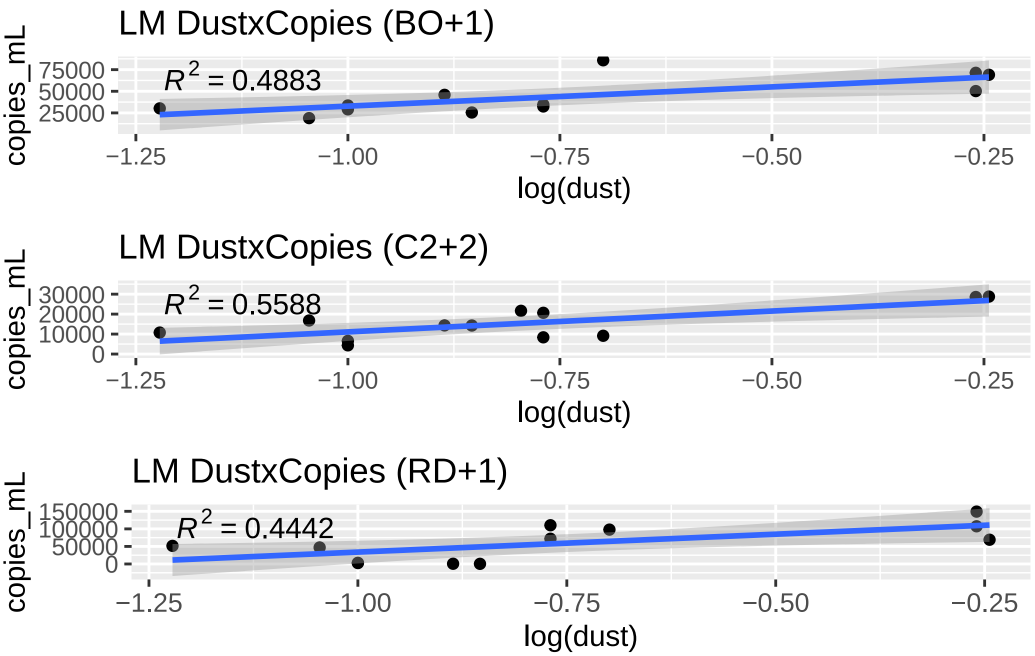
Plot

```
fig1<-
  bo2 %>% ggplot(aes(x = lag, y = copies_mL)) +
  geom_point() +
  stat_smooth(method = "lm")+
  stat_poly_eq(aes(x = lag, y = copies_mL),
               rr.digits = 4)+
  labs(title = "LM DustxCopies (BO+1)",
       x = "log(dust)")
 fig2<-
   c22 %>% ggplot(aes(x = lag, y = copies_mL)) +
  geom_point() +
  stat_smooth(method = "lm")+
  stat_poly_eq(aes(x = lag, y = copies_mL),
               rr.digits = 4)+
  labs(title = "LM DustxCopies (C2+2)",
       x = "log(dust)")
fig3<-
  rd2 %>% ggplot(aes(lag, copies_mL)) +
  geom_point() +
  stat_smooth(method = "lm")+
  stat_poly_eq(aes(x = lag, y = copies_mL),
               rr.digits = 4)+
  labs(title = "LM DustxCopies (RD+1)",
       x = "log(dust)")
```

Stac

```
ggarrange(fig1, fig2, fig3 + font("x.text", size = 10), ncol = 1, nrow = 3)
```

```
`geom_smooth()` using formula = 'y ~ x'
`geom_smooth()` using formula = 'y ~ x'
`geom_smooth()` using formula = 'y ~ x'
```



Null Model:

5-Fold Cross Validation

```
fold_bo <- vfold_cv(bo2, v = 5, repeats = 5) #Data was too small to stratify (<20)
fold_c2 <- vfold_cv(c22, v = 5, repeats = 5) #Data was too small to stratify (<20)
fold_rd <- vfold_cv(rd2, v = 5, repeats = 5) #Data was too small to stratify (<20)
```

Create Null Model

```
#Create Null Recipe for Growth at all three sites
null_bo<- recipe(copies_mL ~ 1, data = bo2)
null_c2<- recipe(copies_mL ~ 1, data = c22)
```

```
null_rd<- recipe(copies_mL ~ 1, data = rd2)

#Set up linear model
lm_mod<- linear_reg() %>%
  set_engine("lm") %>%
  set_mode("regression")

#Workflow that adds recipe to model for three sites
null_wflow_bo<-
  workflow() %>%
  add_model(lm_mod) %>%
  add_recipe(null_bo)

null_wflow_c2<-
  workflow() %>%
  add_model(lm_mod) %>%
  add_recipe(null_c2)

null_wflow_rd<-
  workflow() %>%
  add_model(lm_mod) %>%
  add_recipe(null_rd)

#Use workflow to fit model to  data set
null_fit_bo<-
  fit_resamples(null_wflow_bo, resamples = fold_bo)
```

```
! Fold1, Repeat1: internal: A correlation computation is required, but `estimate` is constan

! Fold2, Repeat1: internal: A correlation computation is required, but `estimate` is constan

! Fold3, Repeat1: internal: A correlation computation is required, but `estimate` is constan

! Fold4, Repeat1: internal: A correlation computation is required, but `estimate` is constan

! Fold5, Repeat1: internal: A correlation computation is required, but `estimate` is constan

! Fold1, Repeat2: internal: A correlation computation is required, but `estimate` is constan

! Fold2, Repeat2: internal: A correlation computation is required, but `estimate` is constan
! Fold3, Repeat2: internal: A correlation computation is required, but `estimate` is constan

! Fold4, Repeat2: internal: A correlation computation is required, but `estimate` is constan

! Fold5, Repeat2: internal: A correlation computation is required, but `estimate` is constan

! Fold1, Repeat3: internal: A correlation computation is required, but `estimate` is constan

! Fold2, Repeat3: internal: A correlation computation is required, but `estimate` is constan

! Fold3, Repeat3: internal: A correlation computation is required, but `estimate` is constan

! Fold4, Repeat3: internal: A correlation computation is required, but `estimate` is constan

! Fold5, Repeat3: internal: A correlation computation is required, but `estimate` is constan

! Fold1, Repeat4: internal: A correlation computation is required, but `estimate` is constan

! Fold2, Repeat4: internal: A correlation computation is required, but `estimate` is constan

! Fold3, Repeat4: internal: A correlation computation is required, but `estimate` is constan

! Fold4, Repeat4: internal: A correlation computation is required, but `estimate` is constan

! Fold5, Repeat4: internal: A correlation computation is required, but `estimate` is constan

! Fold1, Repeat5: internal: A correlation computation is required, but `estimate` is constan

! Fold2, Repeat5: internal: A correlation computation is required, but `estimate` is constan

! Fold3, Repeat5: internal: A correlation computation is required, but `estimate` is constan

! Fold4, Repeat5: internal: A correlation computation is required, but `estimate` is constan

! Fold5, Repeat5: internal: A correlation computation is required, but `estimate` is constan
```

```r
null_fit_c2<-
  fit_resamples(null_wflow_c2, resamples = fold_c2)
```

```
! Fold1, Repeat1: internal: A correlation computation is required, but `estimate` is constan

! Fold2, Repeat1: internal: A correlation computation is required, but `estimate` is constan

! Fold3, Repeat1: internal: A correlation computation is required, but `estimate` is constan

! Fold4, Repeat1: internal: A correlation computation is required, but `estimate` is constan

! Fold5, Repeat1: internal: A correlation computation is required, but `estimate` is constan

! Fold1, Repeat2: internal: A correlation computation is required, but `estimate` is constan

! Fold2, Repeat2: internal: A correlation computation is required, but `estimate` is constan

! Fold3, Repeat2: internal: A correlation computation is required, but `estimate` is constan

! Fold4, Repeat2: internal: A correlation computation is required, but `estimate` is constan

! Fold5, Repeat2: internal: A correlation computation is required, but `estimate` is constan

! Fold1, Repeat3: internal: A correlation computation is required, but `estimate` is constan

! Fold2, Repeat3: internal: A correlation computation is required, but `estimate` is constan

! Fold3, Repeat3: internal: A correlation computation is required, but `estimate` is constan

! Fold4, Repeat3: internal: A correlation computation is required, but `estimate` is constan

! Fold5, Repeat3: internal: A correlation computation is required, but `estimate` is constan

! Fold1, Repeat4: internal: A correlation computation is required, but `estimate` is constan

! Fold2, Repeat4: internal: A correlation computation is required, but `estimate` is constan
! Fold3, Repeat4: internal: A correlation computation is required, but `estimate` is constan

! Fold4, Repeat4: internal: A correlation computation is required, but `estimate` is constan

! Fold5, Repeat4: internal: A correlation computation is required, but `estimate` is constan

! Fold1, Repeat5: internal: A correlation computation is required, but `estimate` is constan

! Fold2, Repeat5: internal: A correlation computation is required, but `estimate` is constan

! Fold3, Repeat5: internal: A correlation computation is required, but `estimate` is constan

! Fold4, Repeat5: internal: A correlation computation is required, but `estimate` is constan

! Fold5, Repeat5: internal: A correlation computation is required, but `estimate` is constan
```

```r
null_fit_rd<-
  fit_resamples(null_wflow_rd, resamples = fold_rd)
```

```
! Fold1, Repeat1: internal: A correlation computation is required, but `estimate` is constan

! Fold2, Repeat1: internal: A correlation computation is required, but `estimate` is constan

! Fold3, Repeat1: internal: A correlation computation is required, but `estimate` is constan

! Fold4, Repeat1: internal: A correlation computation is required, but `estimate` is constan

! Fold5, Repeat1: internal: A correlation computation is required, but `estimate` is constan

! Fold1, Repeat2: internal: A correlation computation is required, but `estimate` is constan

! Fold2, Repeat2: internal: A correlation computation is required, but `estimate` is constan

! Fold3, Repeat2: internal: A correlation computation is required, but `estimate` is constan

! Fold4, Repeat2: internal: A correlation computation is required, but `estimate` is constan
! Fold5, Repeat2: internal: A correlation computation is required, but `estimate` is constant

! Fold1, Repeat3: internal: A correlation computation is required, but `estimate` is constant

! Fold2, Repeat3: internal: A correlation computation is required, but `estimate` is constant

! Fold3, Repeat3: internal: A correlation computation is required, but `estimate` is constant

! Fold4, Repeat3: internal: A correlation computation is required, but `estimate` is constant

! Fold5, Repeat3: internal: A correlation computation is required, but `estimate` is constant

! Fold1, Repeat4: internal: A correlation computation is required, but `estimate` is constant

! Fold2, Repeat4: internal: A correlation computation is required, but `estimate` is constant

! Fold3, Repeat4: internal: A correlation computation is required, but `estimate` is constant

! Fold4, Repeat4: internal: A correlation computation is required, but `estimate` is constant

! Fold5, Repeat4: internal: A correlation computation is required, but `estimate` is constant

! Fold1, Repeat5: internal: A correlation computation is required, but `estimate` is constant

! Fold2, Repeat5: internal: A correlation computation is required, but `estimate` is constant

! Fold3, Repeat5: internal: A correlation computation is required, but `estimate` is constant

! Fold4, Repeat5: internal: A correlation computation is required, but `estimate` is constant

! Fold5, Repeat5: internal: A correlation computation is required, but `estimate` is constant
```

Get Metrics for Null

RMSE

```
null_met_bo <-
  collect_metrics(null_fit_bo)

null_met_c2 <-
  collect_metrics(null_fit_c2)

null_met_rd <-
  collect_metrics(null_fit_rd)

null_met_bo
```

```
# A tibble: 2 x 6
  .metric .estimator   mean     n std_err .config
  <chr>   <chr>        <dbl> <int>   <dbl> <chr>
1 rmse    standard   20995.    25   1677. Preprocessor1_Model1
2 rsq     standard     NaN      0     NA  Preprocessor1_Model1
```

```
null_met_c2
```

```
# A tibble: 2 x 6
  .metric .estimator  mean     n std_err .config
  <chr>   <chr>       <dbl> <int>   <dbl> <chr>
1 rmse    standard   8449.    25    475. Preprocessor1_Model1
2 rsq     standard    NaN      0     NA  Preprocessor1_Model1
```

```
null_met_rd
```

```
# A tibble: 2 x 6
  .metric .estimator   mean     n std_err .config
  <chr>   <chr>        <dbl> <int>   <dbl> <chr>
1 rmse    standard   49889.    25   3928. Preprocessor1_Model1
2 rsq     standard     NaN      0     NA  Preprocessor1_Model1
```

```
rmnb<- 20927.53
rmnc<- 7934.59
rmnr<- 49831.58
```

AIC

```r
bo_null_mod<-
  lm(copies_mL ~ 1, data = bo2)
AIC(bo_null_mod)
```

```
[1] 275.9331
```

```r
c2_null_mod<-
  lm(copies_mL ~ 1, data = c22)
AIC(c2_null_mod)
```

```
[1] 253.1876
```

```r
rd_null_mod<-
  lm(copies_mL ~ 1, data = rd2)
AIC(rd_null_mod)
```

```
[1] 296.9821
```

```r
aicnb<- 275.93
aicnc<- 253.19
aicnr<- 296.98
```

We will come back to these metrics once we run the models on our actual data.

Actual Data for Model

Create Linear Model

BLIND OSO:

```r
#Create Recipe for Growth
growth_recipe_bo<- recipe(copies_mL ~ lag, data = bo2)

#Set up linear model
lm_mod<- linear_reg() %>%
  set_engine("lm") %>%
  set_mode("regression")
```

```
#Workflow that adds recipe to model
Growth_wflow<-
  workflow() %>%
  add_model(lm_mod) %>%
  add_recipe(growth_recipe_bo)

#Use workflow to fit model to  data set
growth_fit_bo<- Growth_wflow %>%
  fit(data = bo2)

#View as Tibble
growth_fit_bo %>%
  extract_fit_parsnip() %>%
  tidy()
```

```
# A tibble: 2 x 5
  term        estimate std.error statistic   p.value
  <chr>          <dbl>     <dbl>     <dbl>     <dbl>
1 (Intercept)   77277.    11730.      6.59 0.0000617
2 lag           44471.    14395.      3.09 0.0115
```

CANALS:

```
#Create Recipe for Growth
growth_recipe_c2<- recipe(copies_mL ~ lag, data = c22)

#Set up linear model
lm_mod<- linear_reg() %>%
  set_engine("lm") %>%
  set_mode("regression")

#Workflow that adds recipe to model
Growth_wflow<-
  workflow() %>%
  add_model(lm_mod) %>%
  add_recipe(growth_recipe_c2)

#Use workflow to fit model to  data set
growth_fit_c2<- Growth_wflow %>%
  fit(data = c22)
```

```r
#View as Tibble
growth_fit_c2 %>%
  extract_fit_parsnip() %>%
  tidy()
```

```
# A tibble: 2 x 5
  term        estimate std.error statistic   p.value
  <chr>          <dbl>     <dbl>     <dbl>     <dbl>
1 (Intercept)   31954.     4942.      6.47 0.0000721
2 lag           20859.     5861.      3.56 0.00519
```

GULF:

```r
#Create Recipe for Growth
growth_recipe_rd<- recipe(copies_mL ~ lag, data = rd2)

#Set up linear model
lm_mod<- linear_reg() %>%
  set_engine("lm") %>%
  set_mode("regression")

#Workflow that adds recipe to model
Growth_wflow<-
  workflow() %>%
  add_model(lm_mod) %>%
  add_recipe(growth_recipe_rd)

#Use workflow to fit model to  data set
growth_fit_rd<- Growth_wflow %>%
  fit(data = rd2)

#View as Tibble
growth_fit_rd %>%
  extract_fit_parsnip() %>%
  tidy()
```

```
# A tibble: 2 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)  135834.    29387.      4.62 0.000947
```

```
2 lag           101960.    36063.      2.83 0.0179
```

Metrics

RMSE and R2

```
aug_test_bo <- augment(growth_fit_bo, bo2)
rmse <- aug_test_bo %>% rmse(truth = copies_mL, .pred)
rsq <- aug_test_bo %>% rsq(truth = copies_mL, .pred)
bo_metrics<- full_join(rmse, rsq)
```

```
Joining, by = c(".metric", ".estimator", ".estimate")
```

```
bo_metrics
```

```
# A tibble: 2 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 rmse    standard   14423.
2 rsq     standard       0.488
```

```
aug_test_c2 <- augment(growth_fit_c2, c22)
rmse <- aug_test_c2 %>% rmse(truth = copies_mL, .pred)
rsq <- aug_test_c2 %>% rsq(truth = copies_mL, .pred)
c2_metrics<- full_join(rmse, rsq)
```

```
Joining, by = c(".metric", ".estimator", ".estimate")
```

```
c2_metrics
```

```
# A tibble: 2 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 rmse    standard    5191.
2 rsq     standard       0.559
```

```
aug_test_rd <- augment(growth_fit_rd, rd2)
rmse <- aug_test_rd %>% rmse(truth = copies_mL, .pred)
rsq <- aug_test_rd %>% rsq(truth = copies_mL, .pred)
rd_metrics<- full_join(rmse, rsq)
```

```
Joining, by = c(".metric", ".estimator", ".estimate")
```

```
rd_metrics
```

```
# A tibble: 2 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 rmse    standard   36132.   
2 rsq     standard       0.444
```

```
rmb<- 14422.91
rmc<- 5191.22
rmr<- 36132.11
```

AIC

Approaches based on such selection criteria essentially try to guess how the model would perform if it were to be fit to new data, without actually trying to do it (in contrast to CV). For AIC, a rule of thumb is that a difference of 10 between 2 models is meaningful.

```
bo_mod<- lm(copies_mL ~ lag, data = bo2)
AIC(bo_mod)
```

```
[1] 269.8923
```

```
c2_mod<- lm(copies_mL ~ lag, data = c22)
AIC(c2_mod)
```

```
[1] 245.3679
```

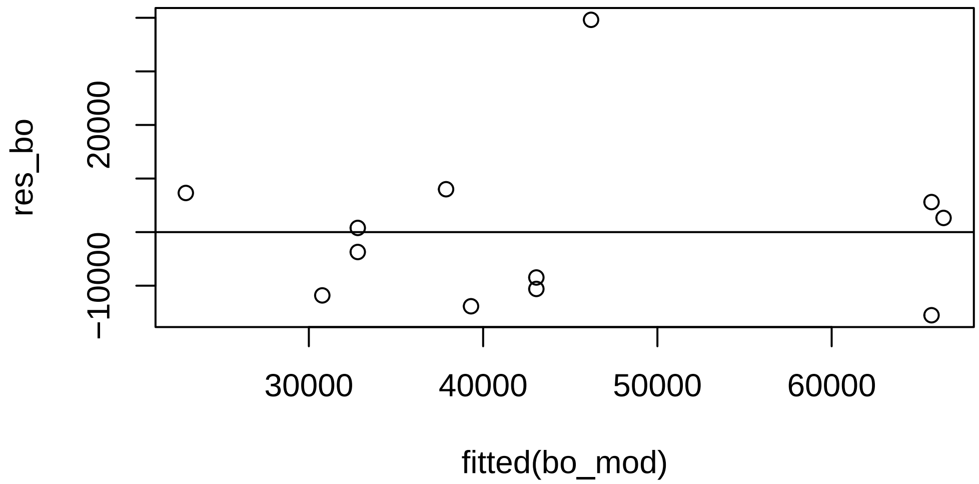
```
rd_mod<- lm(copies_mL ~ lag, data = rd2)
AIC(rd_mod)
```

```
[1] 291.933
```

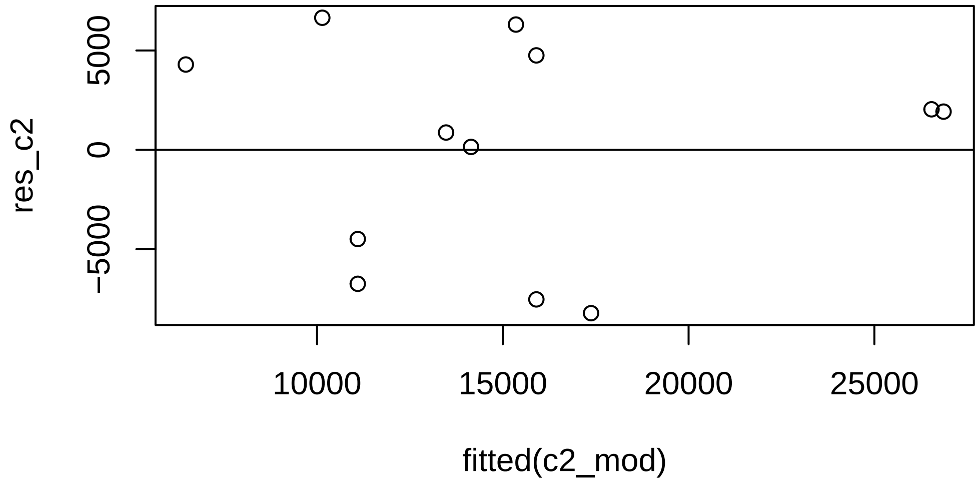
```
aicb<- 269.89
aicc<- 245.37
aicr<- 291.93
```

Plot Residuals

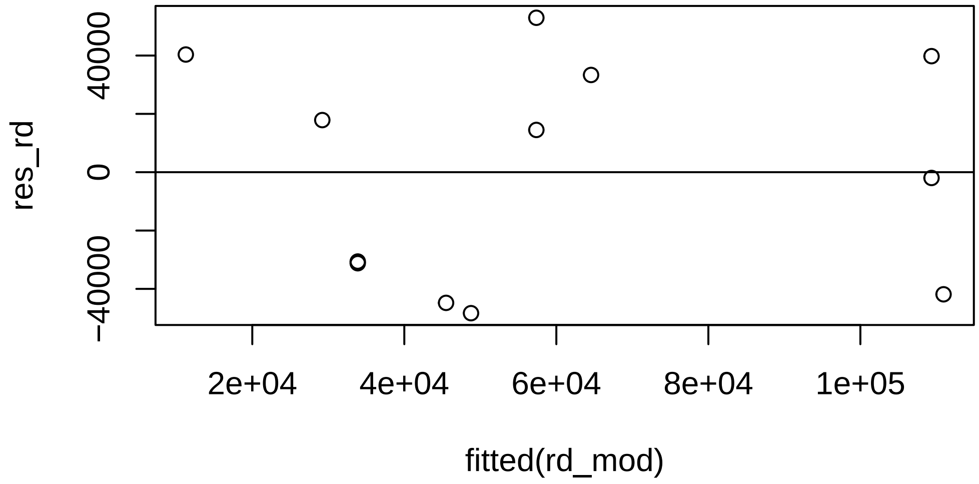
```
res_bo<- resid(bo_mod)
plot(fitted(bo_mod), res_bo)
abline(0,0)
```



```
res_c2<- resid(c2_mod)
plot(fitted(c2_mod), res_c2)
abline(0,0)
```



```
res_rd<- resid(rd_mod)
plot(fitted(rd_mod), res_rd)
abline(0,0)
```



Conclusion

Site	Model Type	RMSE	AIC
BO	Null	20472.82	275.9331
BO	Actual	14422.91	269.8923
C2	Null	8472.898	253.1876
C2	Actual	5191.22	245.3679
RD	Null	50386.56	296.9821
RD	Actual	36132.11	291.933

AIC and RMSE are lower (better performance) than the null.

Make Table

```
kable(data.frame(Model = c("Null RMSE", "Actual RMSE", "Null AIC ", "Actual AIC"),
                 Blind_Oso = c(rmnb,rmb,aicnb, aicb),
                 Canals = c(rmnc, rmc, aicnc, aicc),
                 Gulf = c(rmnr, rmr, aicnr, aicr)),
      caption = "Linear Regression Model Metrics")
```

Table 2: Linear Regression Model Metrics

Model	Blind_Oso	Canals	Gulf
Null RMSE	20927.53	7934.59	49831.58
Actual RMSE	14422.91	5191.22	36132.11
Null AIC	275.93	253.19	296.98
Actual AIC	269.89	245.37	291.93

OTHER ENVIRONMENTAL PARAMETERS

Create New Master Data set

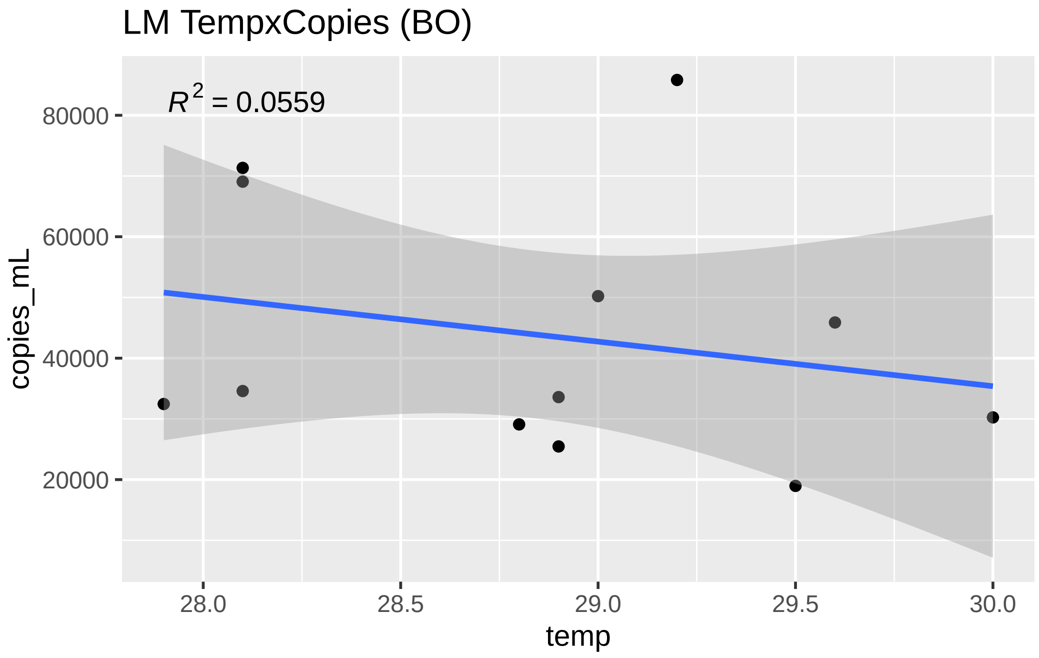
```
master2<-
  master0 %>%
  select(copies_mL,sal,temp,site, amm, orthop, doc, don, tn, do_mgl, nn, sil, din_dip, tdn

#Make Site Specific DF
b<-
  master2 %>%
  filter(site %in% "Blind Oso")

c<-
  master2 %>%
  filter(site %in% "Canals")

r<-
  master2 %>%
  filter(site %in% "Gulf")
```

Plot

Temperature

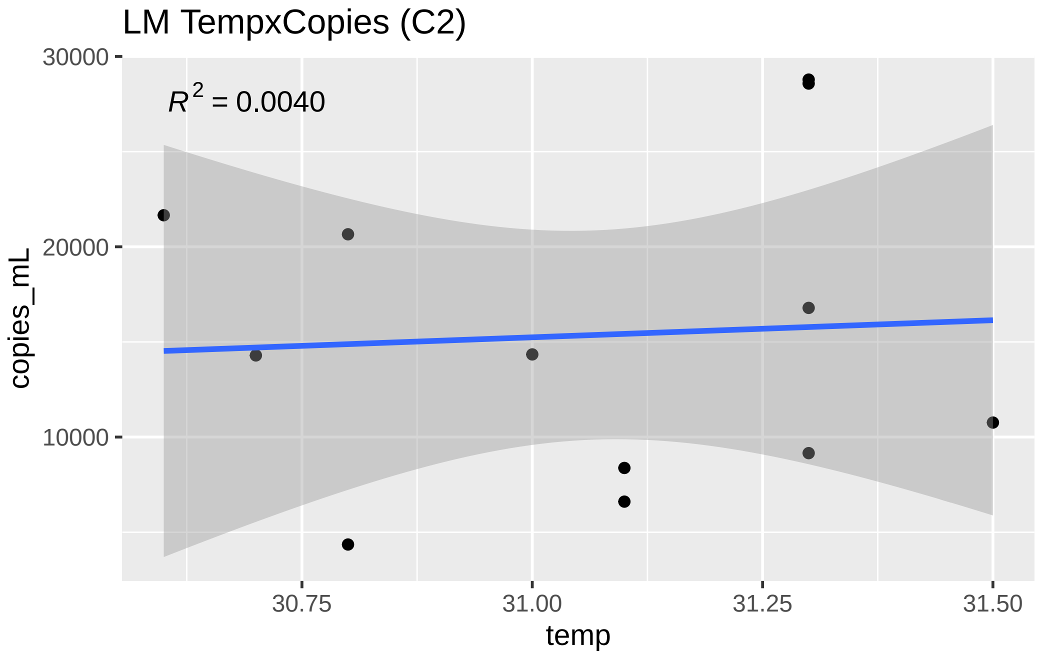
```
b %>% ggplot(aes(x = temp, y = copies_mL)) +
  geom_point() +
  stat_smooth(method = "lm")+
  stat_poly_eq(aes(x = temp, y = copies_mL),
               rr.digits = 4) +
  labs(title = "LM TempxCopies (BO)")
```

```
`geom_smooth()` using formula = 'y ~ x'
```



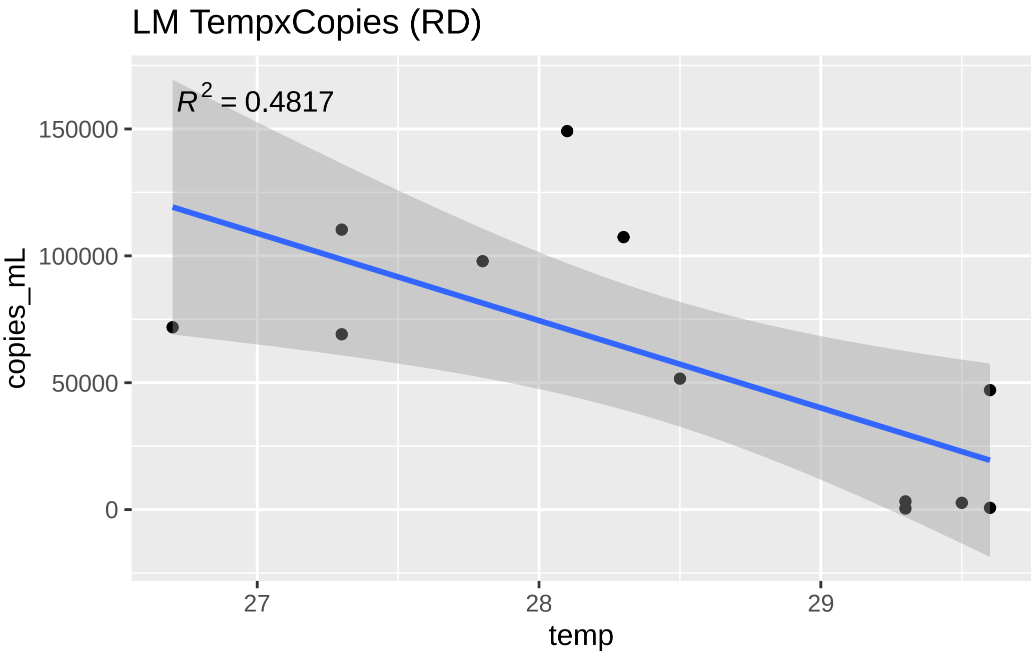
```r
c %>% ggplot(aes(x = temp, y = copies_mL)) +
  geom_point() +
  stat_smooth(method = "lm")+
  stat_poly_eq(aes(x = temp, y = copies_mL),
               rr.digits = 4) +
  labs(title = "LM TempxCopies (C2)")
```

```
`geom_smooth()` using formula = 'y ~ x'
```



```
r %>% ggplot(aes(x = temp, y = copies_mL)) +
  geom_point() +
  stat_smooth(method = "lm")+
  stat_poly_eq(aes(x = temp, y = copies_mL),
               rr.digits = 4) +
  labs(title = "LM TempxCopies (RD)")
```

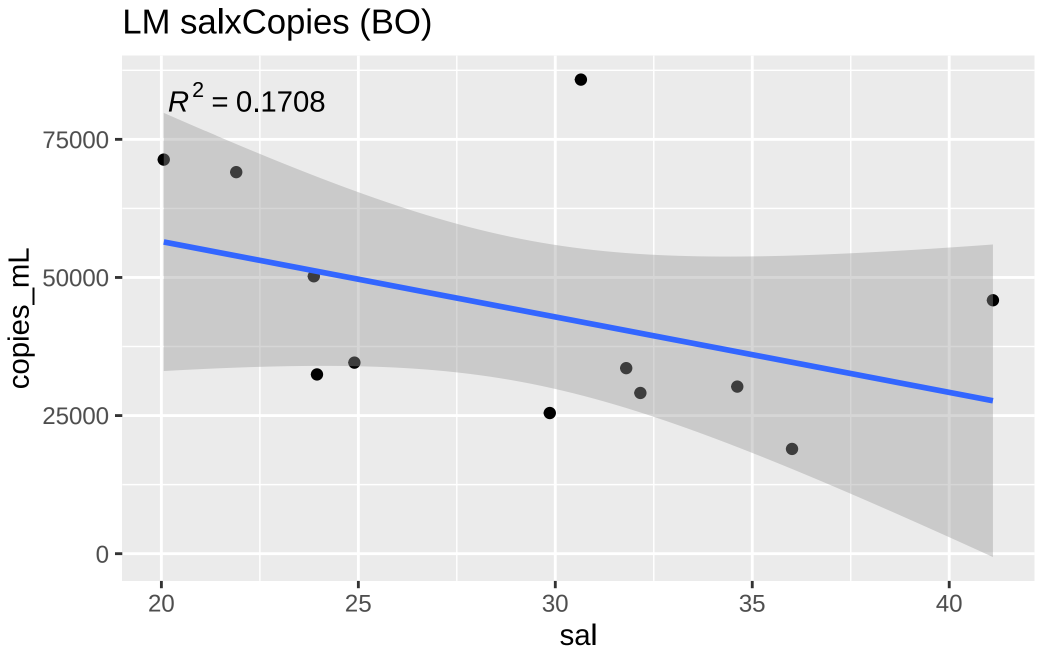
```
`geom_smooth()` using formula = 'y ~ x'
```



Salinity

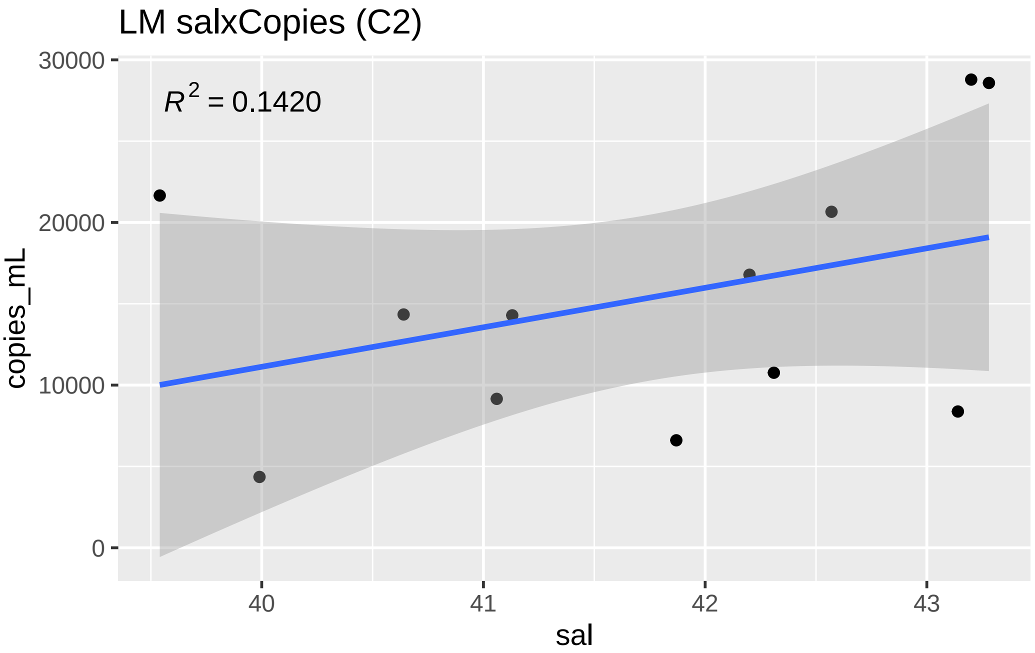
```
b %>% ggplot(aes(x = sal, y = copies_mL)) +
  geom_point() +
  stat_smooth(method = "lm")+
  stat_poly_eq(aes(x = sal, y = copies_mL),
               rr.digits = 4) +
  labs(title = "LM salxCopies (BO)")
```

```
`geom_smooth()` using formula = 'y ~ x'
```



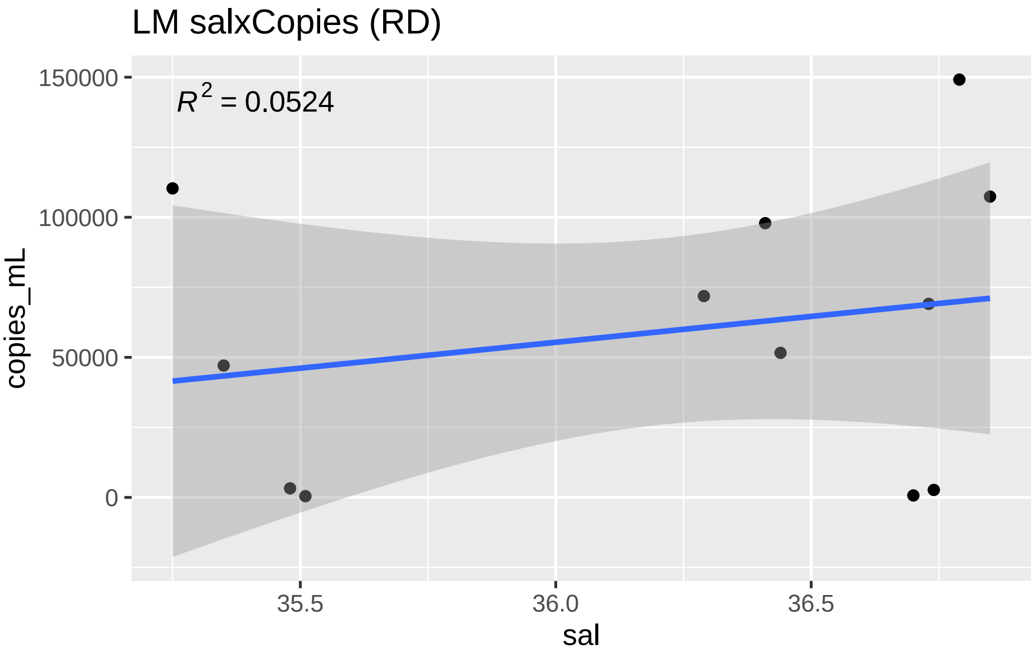
```
c %>% ggplot(aes(x = sal, y = copies_mL)) +
  geom_point() +
  stat_smooth(method = "lm")+
  stat_poly_eq(aes(x = sal, y = copies_mL),
               rr.digits = 4) +
  labs(title = "LM salxCopies (C2)")
```

```
`geom_smooth()` using formula = 'y ~ x'
```



```
r %>% ggplot(aes(x = sal, y = copies_mL)) +
  geom_point() +
  stat_smooth(method = "lm")+
  stat_poly_eq(aes(x = sal, y = copies_mL),
               rr.digits = 4) +
  labs(title = "LM salxCopies (RD)")
```

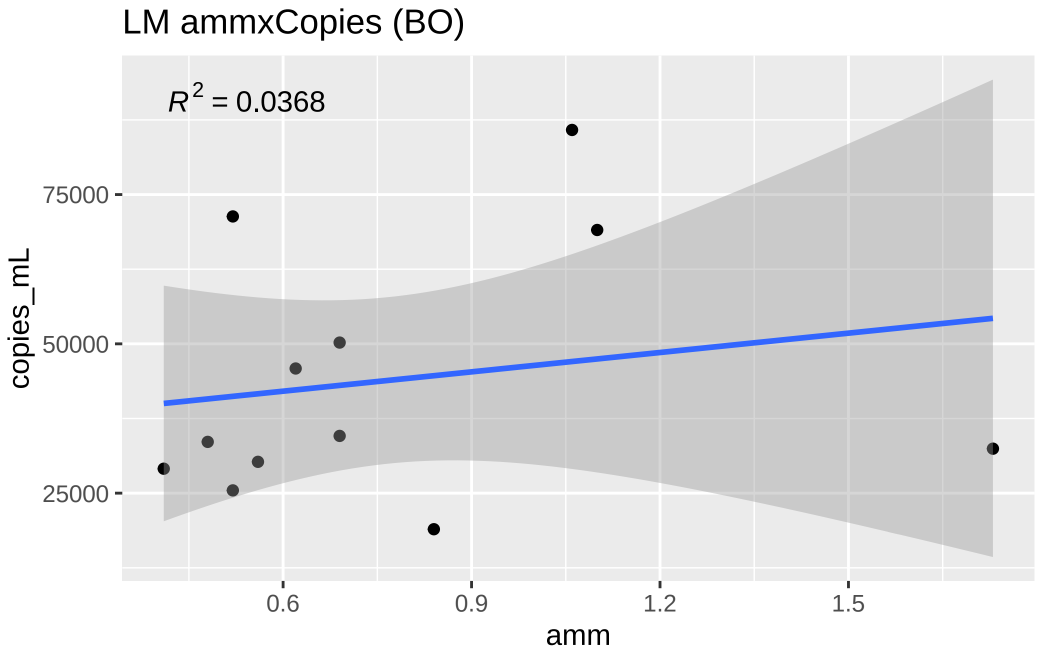
```
`geom_smooth()` using formula = 'y ~ x'
```



Ammonia

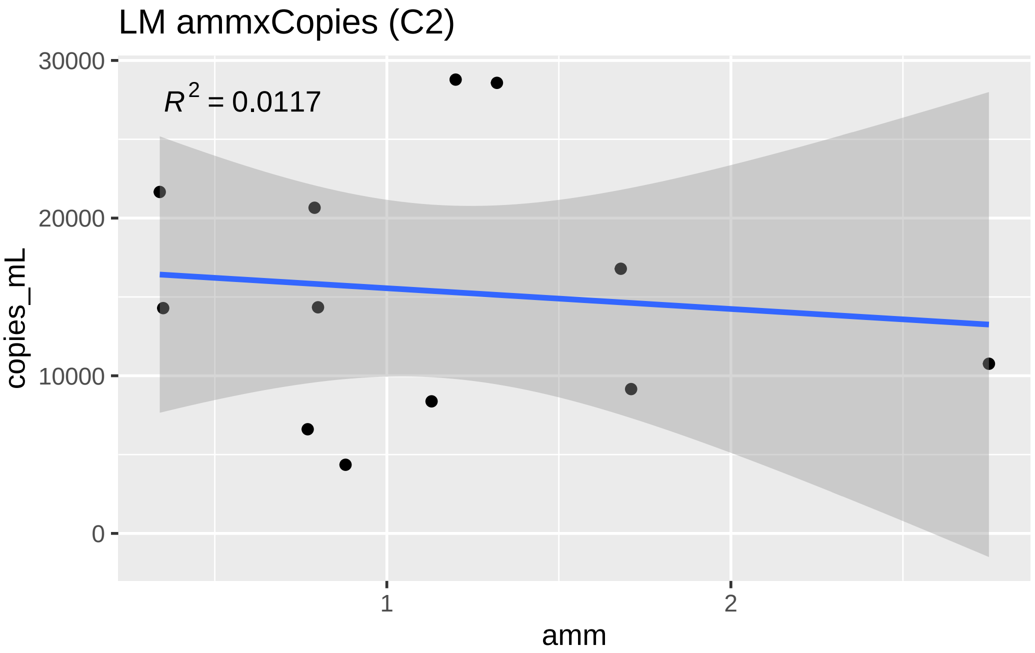
```
b %>% ggplot(aes(x = amm, y = copies_mL)) +
  geom_point() +
  stat_smooth(method = "lm")+
  stat_poly_eq(aes(x = amm, y = copies_mL),
               rr.digits = 4) +
  labs(title = "LM ammxCopies (BO)")
```

```
`geom_smooth()` using formula = 'y ~ x'
```



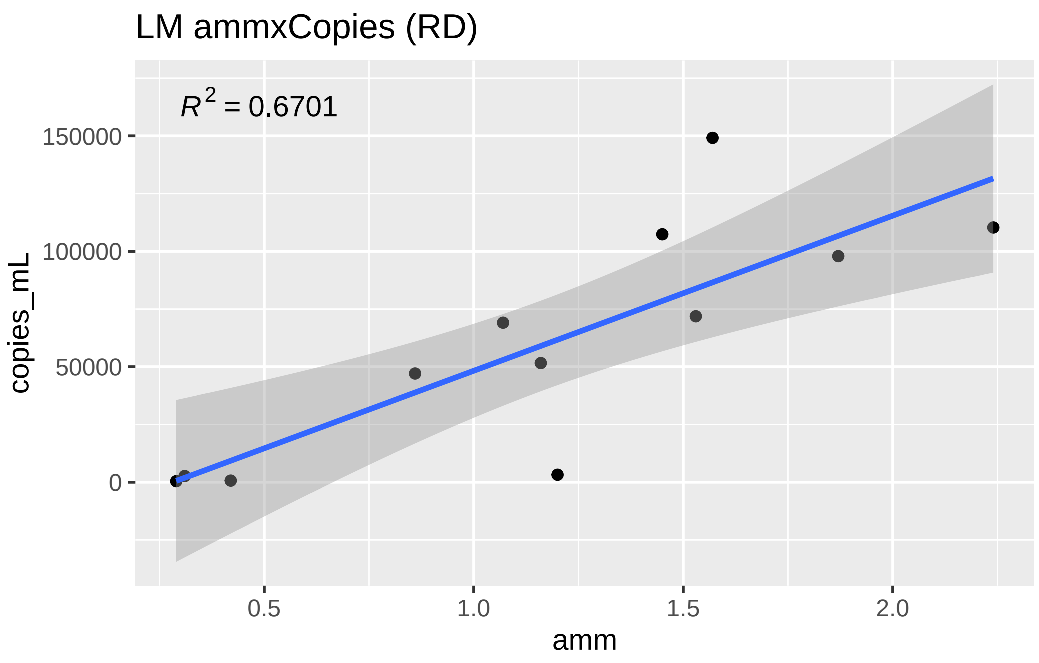
```
c %>% ggplot(aes(x = amm, y = copies_mL)) +
  geom_point() +
  stat_smooth(method = "lm")+
  stat_poly_eq(aes(x = amm, y = copies_mL),
               rr.digits = 4) +
  labs(title = "LM ammxCopies (C2)")
```

```
`geom_smooth()` using formula = 'y ~ x'
```



```
r %>% ggplot(aes(x = amm, y = copies_mL)) +
  geom_point() +
  stat_smooth(method = "lm")+
  stat_poly_eq(aes(x = amm, y = copies_mL),
               rr.digits = 4) +
  labs(title = "LM ammxCopies (RD)")
```

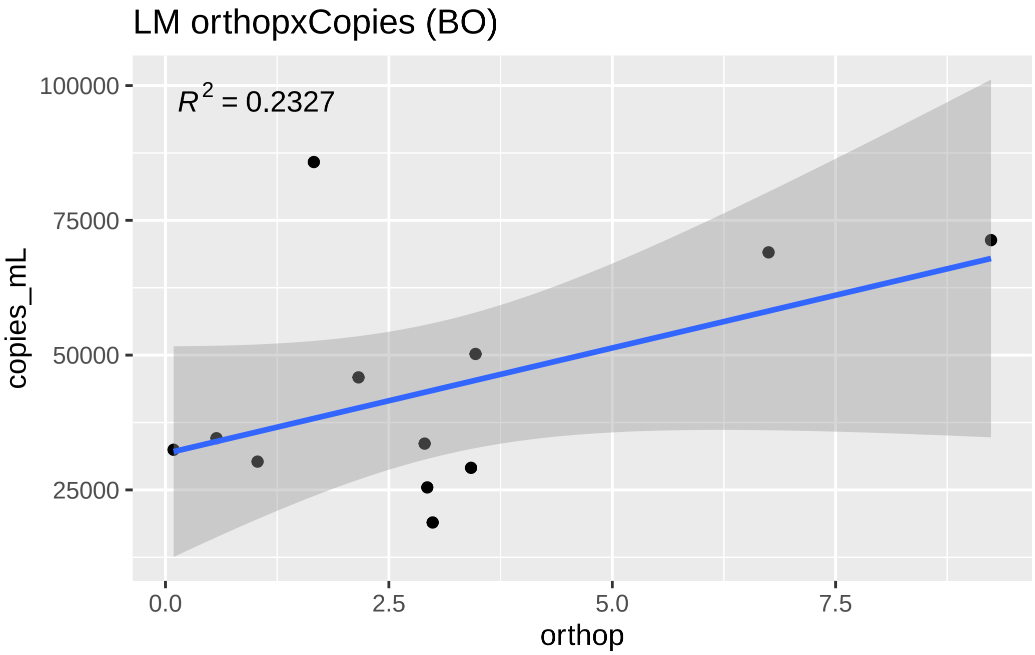
```
`geom_smooth()` using formula = 'y ~ x'
```



Orthophosphate

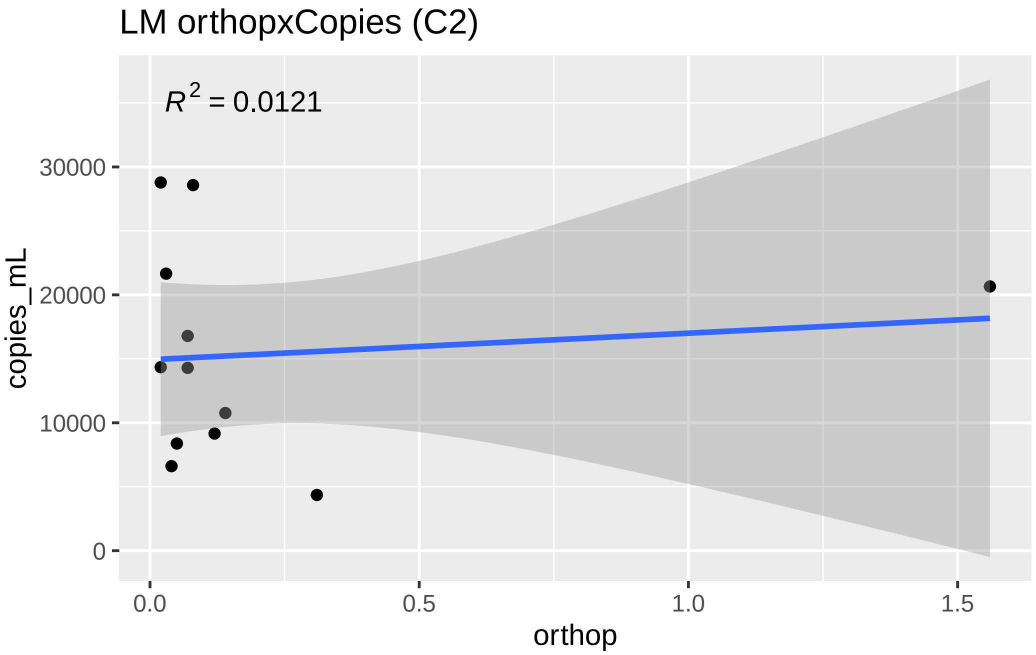
```
b %>% ggplot(aes(x = orthop, y = copies_mL)) +
  geom_point() +
  stat_smooth(method = "lm")+
  stat_poly_eq(aes(x = orthop, y = copies_mL),
               rr.digits = 4) +
  labs(title = "LM orthopxCopies (BO)")
```

```
`geom_smooth()` using formula = 'y ~ x'
```



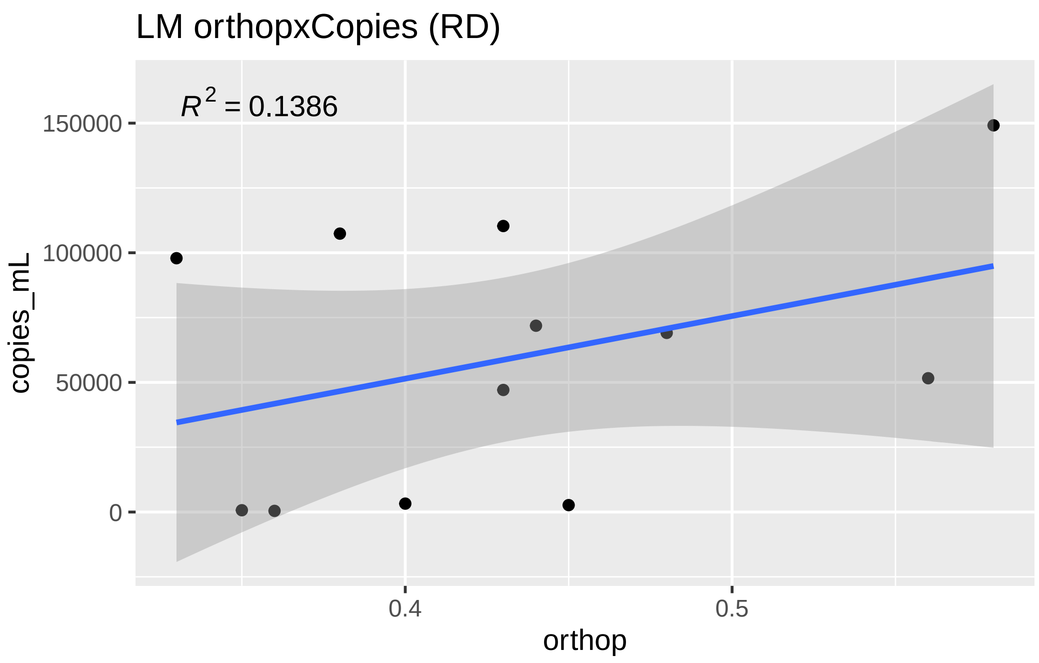
```
c %>% ggplot(aes(x = orthop, y = copies_mL)) +
  geom_point() +
  stat_smooth(method = "lm")+
  stat_poly_eq(aes(x = orthop, y = copies_mL),
               rr.digits = 4) +
  labs(title = "LM orthopxCopies (C2)")
```

```
`geom_smooth()` using formula = 'y ~ x'
```



```
r %>% ggplot(aes(x = orthop, y = copies_mL)) +
  geom_point() +
  stat_smooth(method = "lm")+
  stat_poly_eq(aes(x = orthop, y = copies_mL),
               rr.digits = 4) +
  labs(title = "LM orthopxCopies (RD)")
```

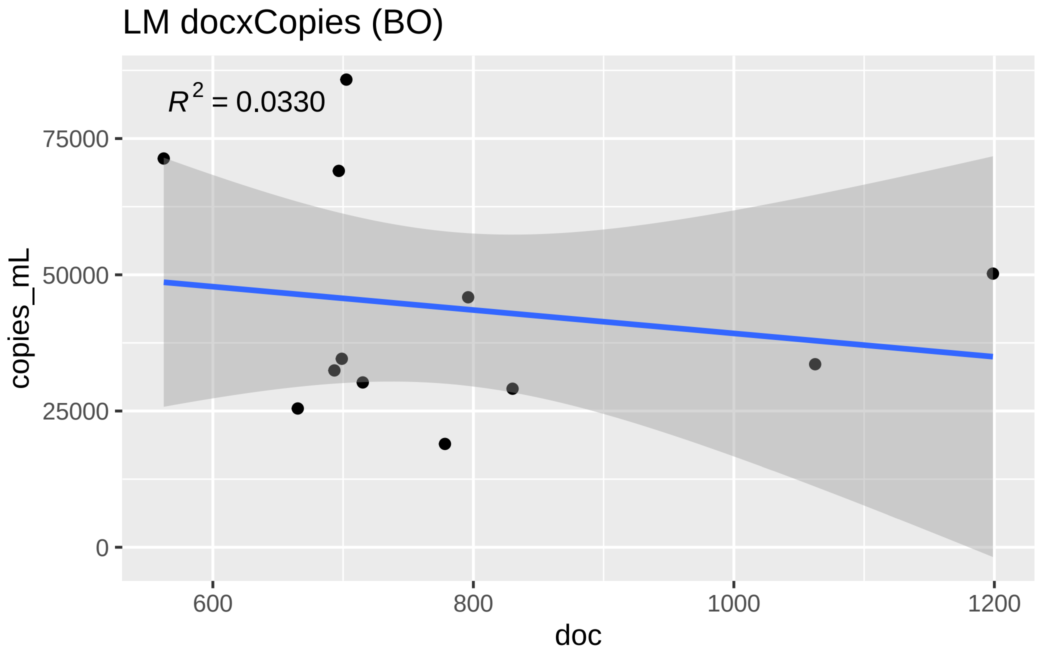
```
`geom_smooth()` using formula = 'y ~ x'
```



DOC

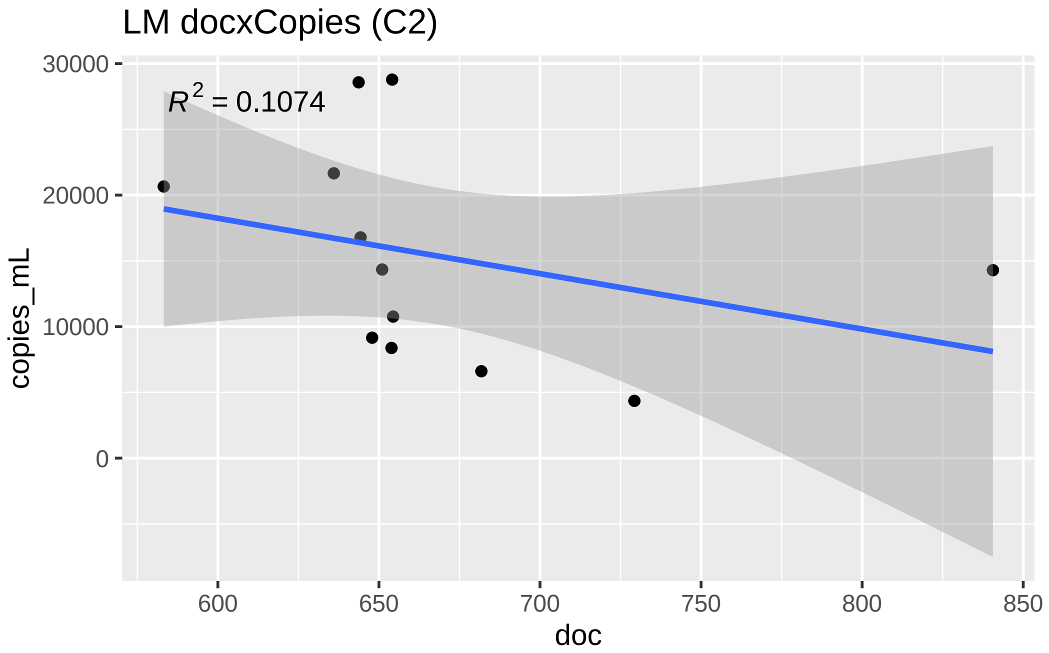
```
b %>% ggplot(aes(x = doc, y = copies_mL)) +
  geom_point() +
  stat_smooth(method = "lm")+
  stat_poly_eq(aes(x = doc, y = copies_mL),
               rr.digits = 4) +
  labs(title = "LM docxCopies (BO)")
```

```
`geom_smooth()` using formula = 'y ~ x'
```



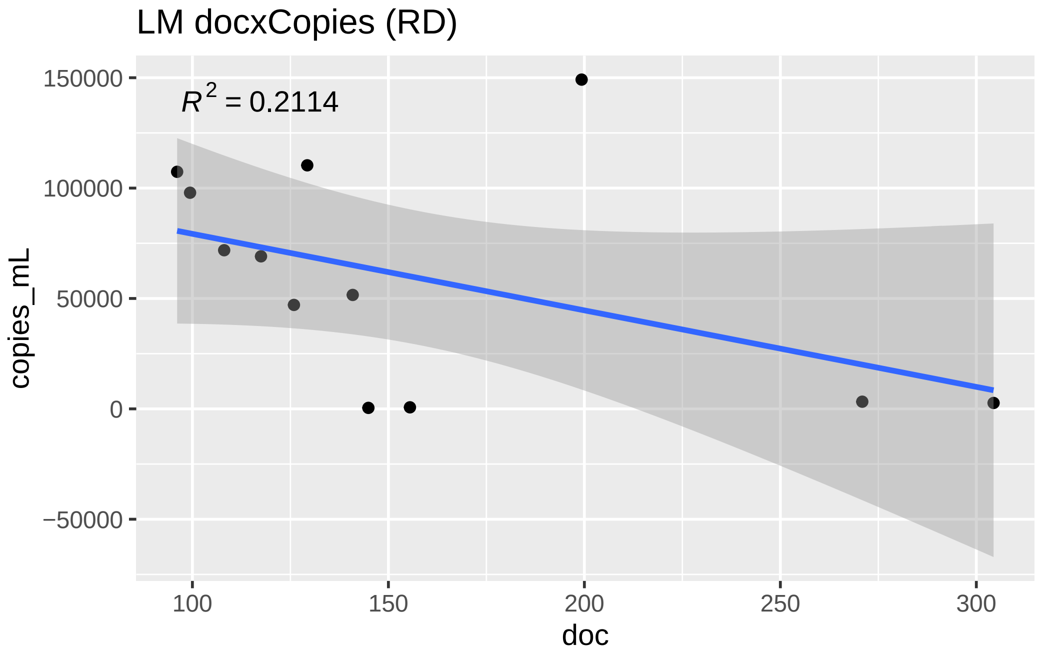
```
c %>% ggplot(aes(x = doc, y = copies_mL)) +
  geom_point() +
  stat_smooth(method = "lm")+
  stat_poly_eq(aes(x = doc, y = copies_mL),
               rr.digits = 4) +
  labs(title = "LM docxCopies (C2)")
```

```
`geom_smooth()` using formula = 'y ~ x'
```



```
r %>% ggplot(aes(x = doc, y = copies_mL)) +
  geom_point() +
  stat_smooth(method = "lm")+
  stat_poly_eq(aes(x = doc, y = copies_mL),
               rr.digits = 4) +
  labs(title = "LM docxCopies (RD)")
```

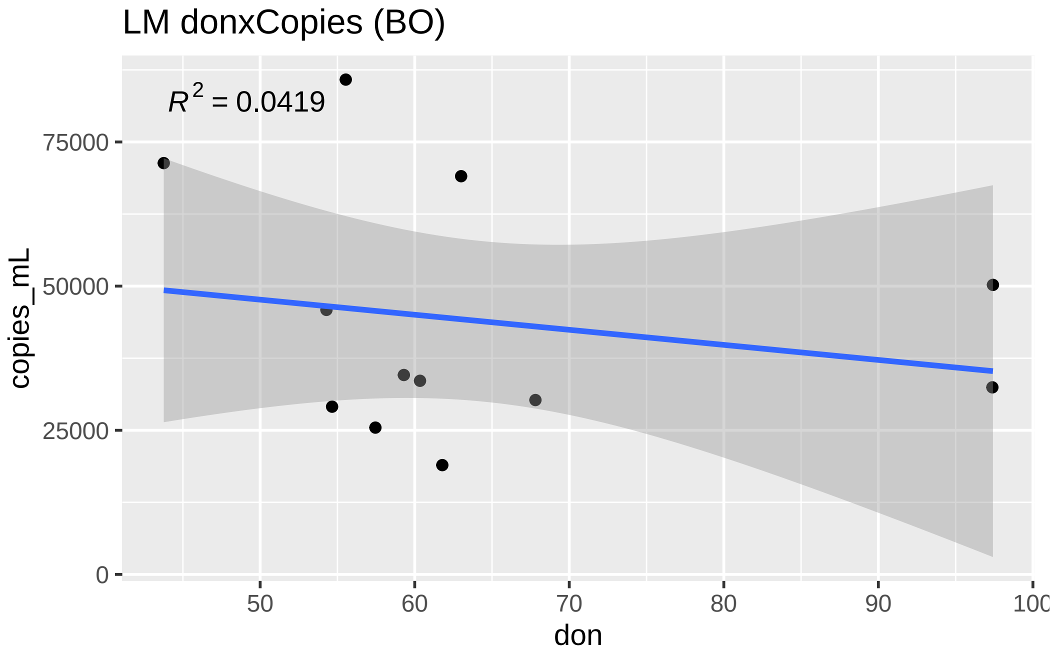
```
`geom_smooth()` using formula = 'y ~ x'
```



DON

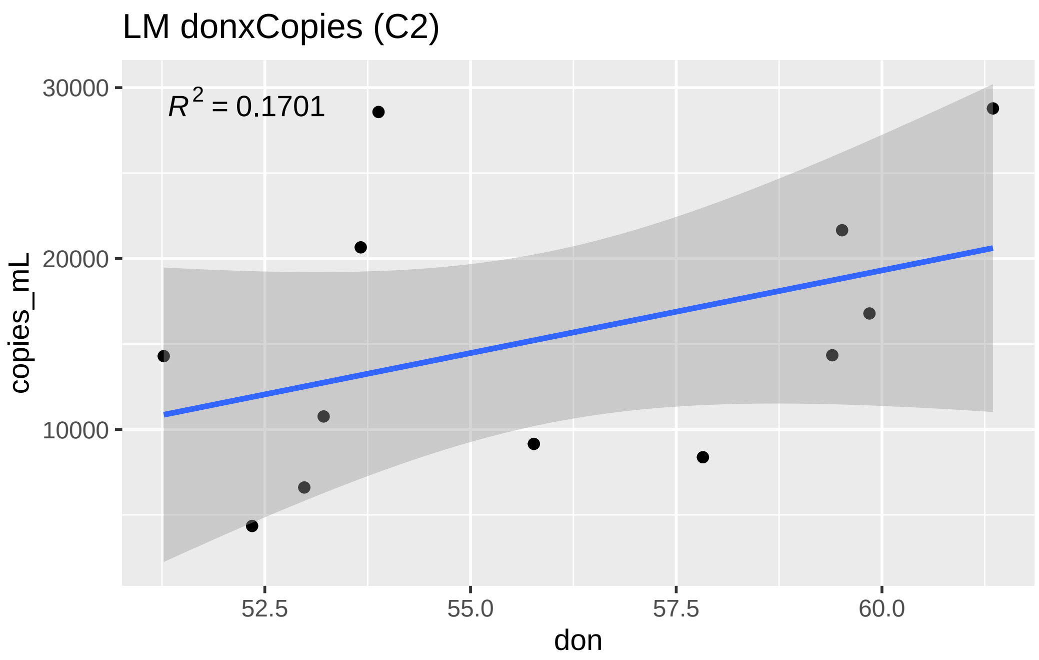
```
b %>% ggplot(aes(x = don, y = copies_mL)) +
  geom_point() +
  stat_smooth(method = "lm")+
  stat_poly_eq(aes(x = don, y = copies_mL),
               rr.digits = 4) +
  labs(title = "LM donxCopies (BO)")
```

```
`geom_smooth()` using formula = 'y ~ x'
```



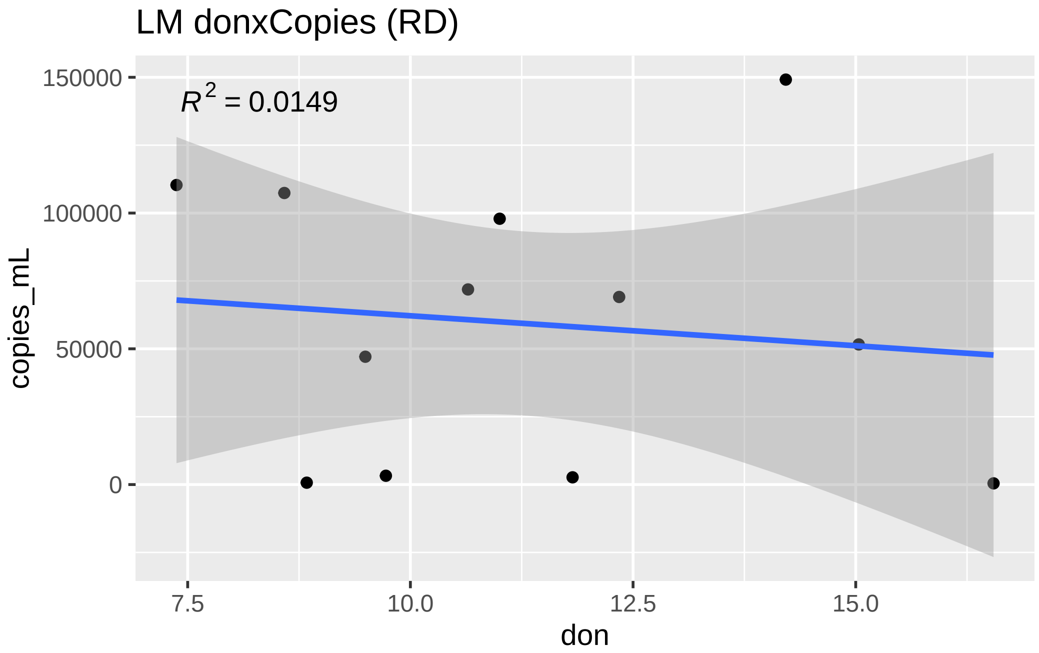
```
c %>% ggplot(aes(x = don, y = copies_mL)) +
  geom_point() +
  stat_smooth(method = "lm")+
  stat_poly_eq(aes(x = don, y = copies_mL),
               rr.digits = 4) +
  labs(title = "LM donxCopies (C2)")
```

```
`geom_smooth()` using formula = 'y ~ x'
```



```
r %>% ggplot(aes(x = don, y = copies_mL)) +
  geom_point() +
  stat_smooth(method = "lm")+
  stat_poly_eq(aes(x = don, y = copies_mL),
               rr.digits = 4) +
  labs(title = "LM donxCopies (RD)")
```

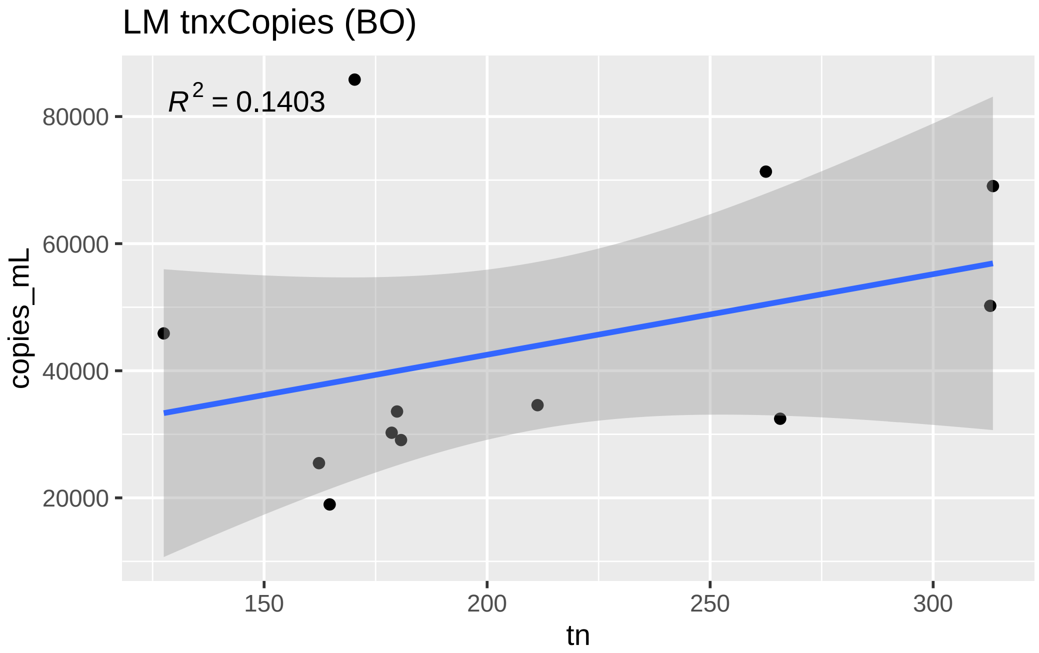
```
`geom_smooth()` using formula = 'y ~ x'
```



Total Nitrogen

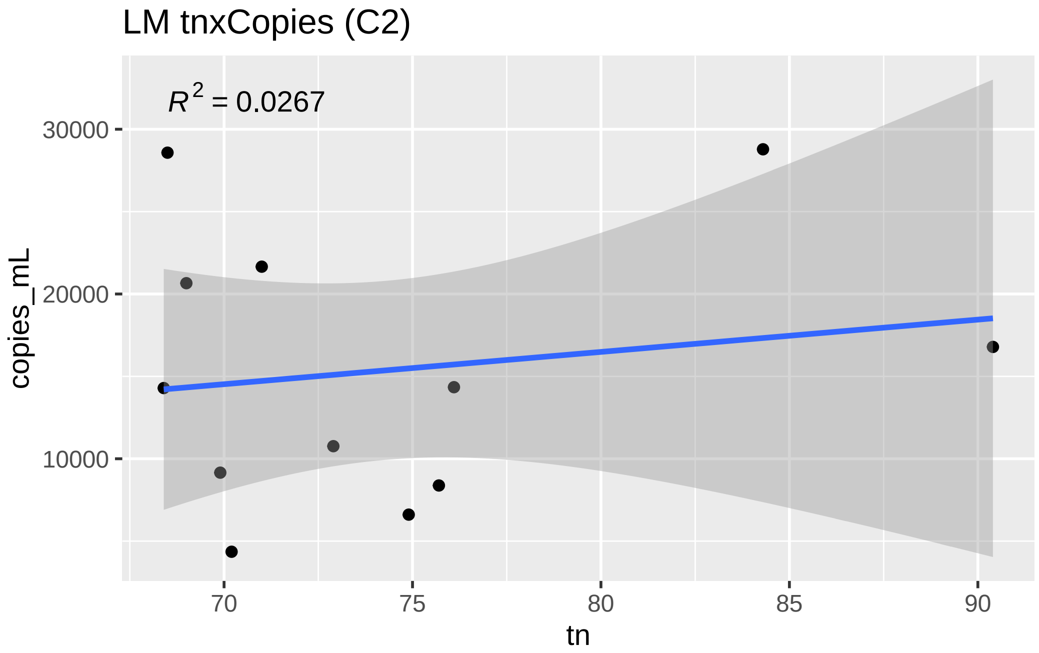
```
b %>% ggplot(aes(x = tn, y = copies_mL)) +
  geom_point() +
  stat_smooth(method = "lm")+
  stat_poly_eq(aes(x = tn, y = copies_mL),
               rr.digits = 4) +
  labs(title = "LM tnxCopies (BO)")
```

```
`geom_smooth()` using formula = 'y ~ x'
```



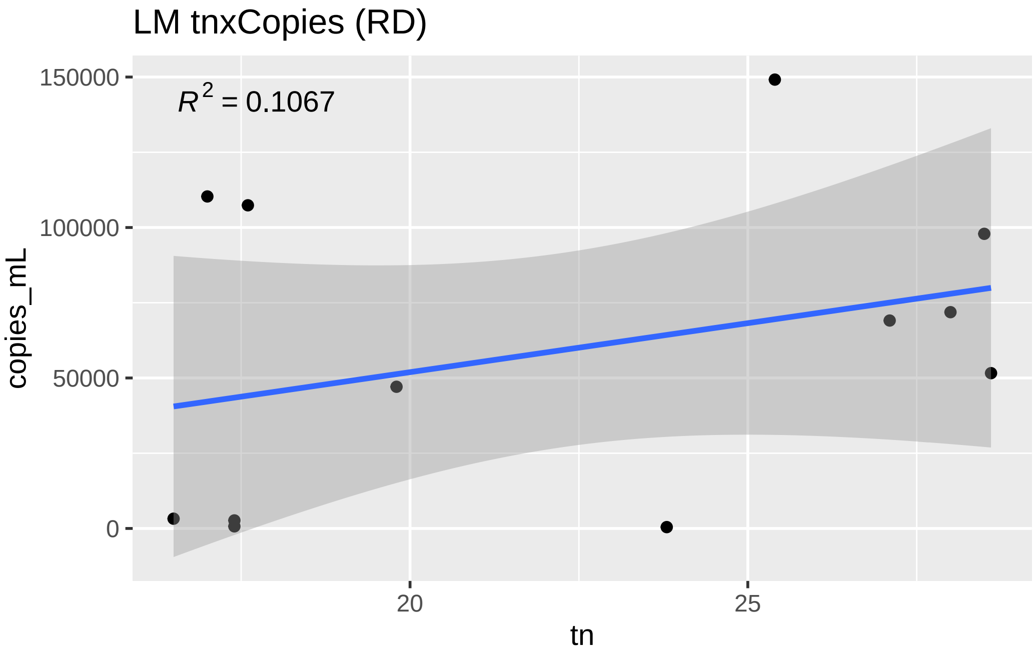
```
c %>% ggplot(aes(x = tn, y = copies_mL)) +
  geom_point() +
  stat_smooth(method = "lm")+
  stat_poly_eq(aes(x = tn, y = copies_mL),
               rr.digits = 4) +
  labs(title = "LM tnxCopies (C2)")
```

```
`geom_smooth()` using formula = 'y ~ x'
```



```
r %>% ggplot(aes(x = tn, y = copies_mL)) +
  geom_point() +
  stat_smooth(method = "lm")+
  stat_poly_eq(aes(x = tn, y = copies_mL),
               rr.digits = 4) +
  labs(title = "LM tnxCopies (RD)")
```

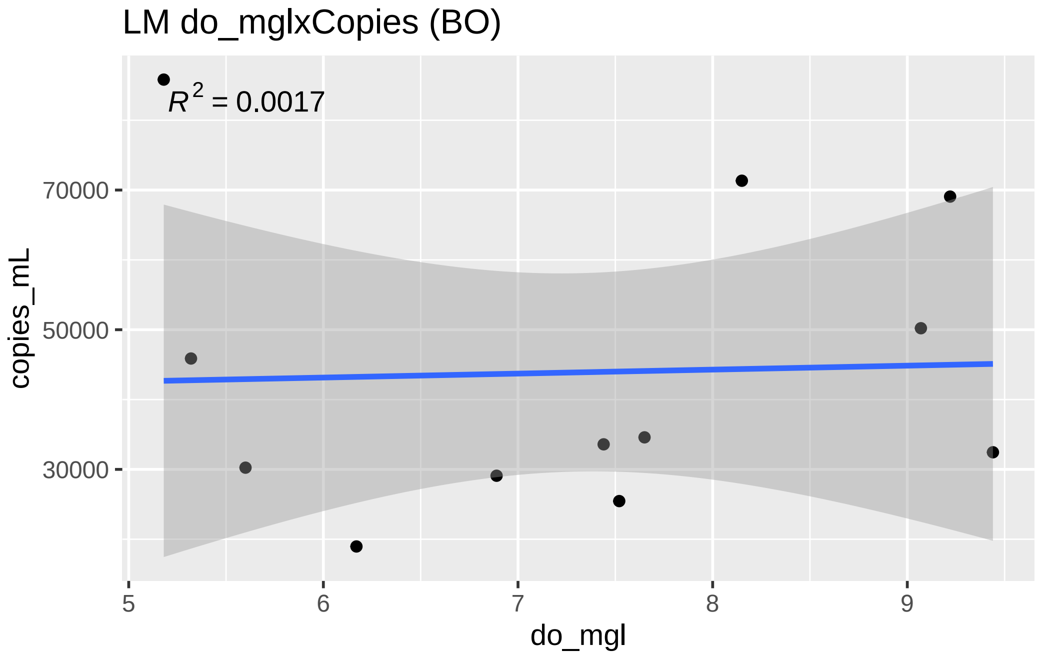
```
`geom_smooth()` using formula = 'y ~ x'
```



DO

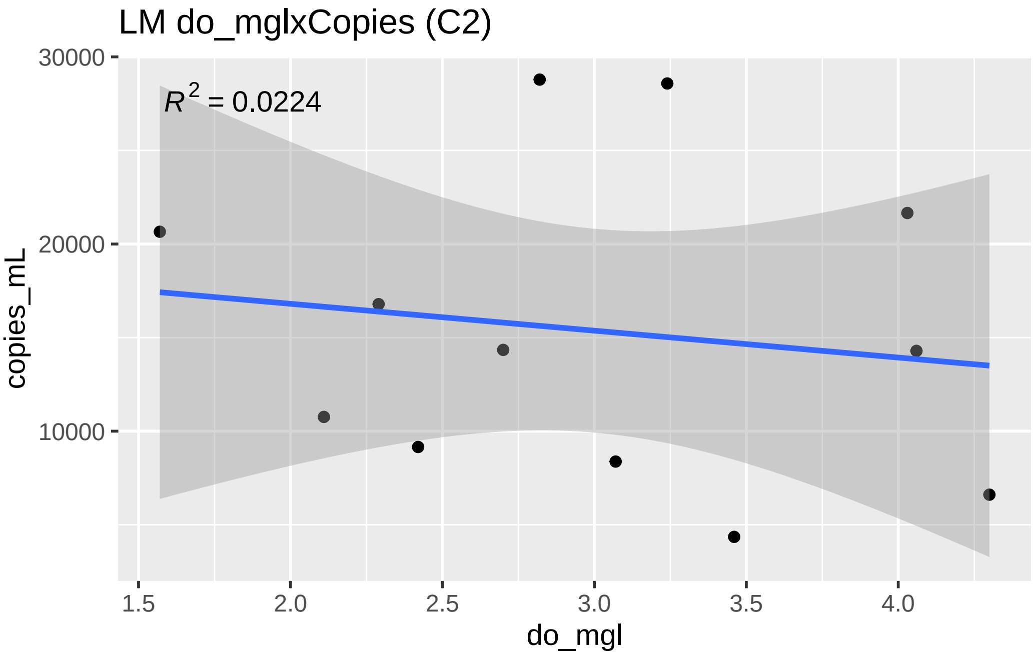
```
b %>% ggplot(aes(x = do_mgl, y = copies_mL)) +
  geom_point() +
  stat_smooth(method = "lm")+
  stat_poly_eq(aes(x = do_mgl, y = copies_mL),
               rr.digits = 4) +
  labs(title = "LM do_mglxCopies (BO)")
```

```
`geom_smooth()` using formula = 'y ~ x'
```



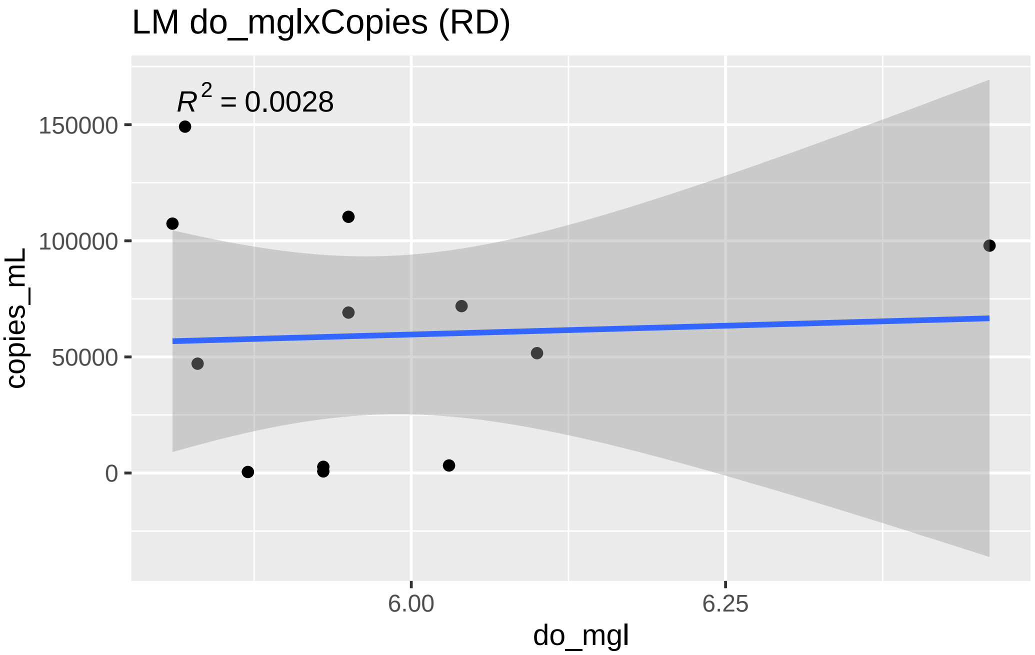
```
c %>% ggplot(aes(x = do_mgl, y = copies_mL)) +
  geom_point() +
  stat_smooth(method = "lm")+
  stat_poly_eq(aes(x = do_mgl, y = copies_mL),
               rr.digits = 4) +
  labs(title = "LM do_mglxCopies (C2)")
```

```
`geom_smooth()` using formula = 'y ~ x'
```



```
r %>% ggplot(aes(x = do_mgl, y = copies_mL)) +
  geom_point() +
  stat_smooth(method = "lm")+
  stat_poly_eq(aes(x = do_mgl, y = copies_mL),
               rr.digits = 4) +
  labs(title = "LM do_mglxCopies (RD)")
```

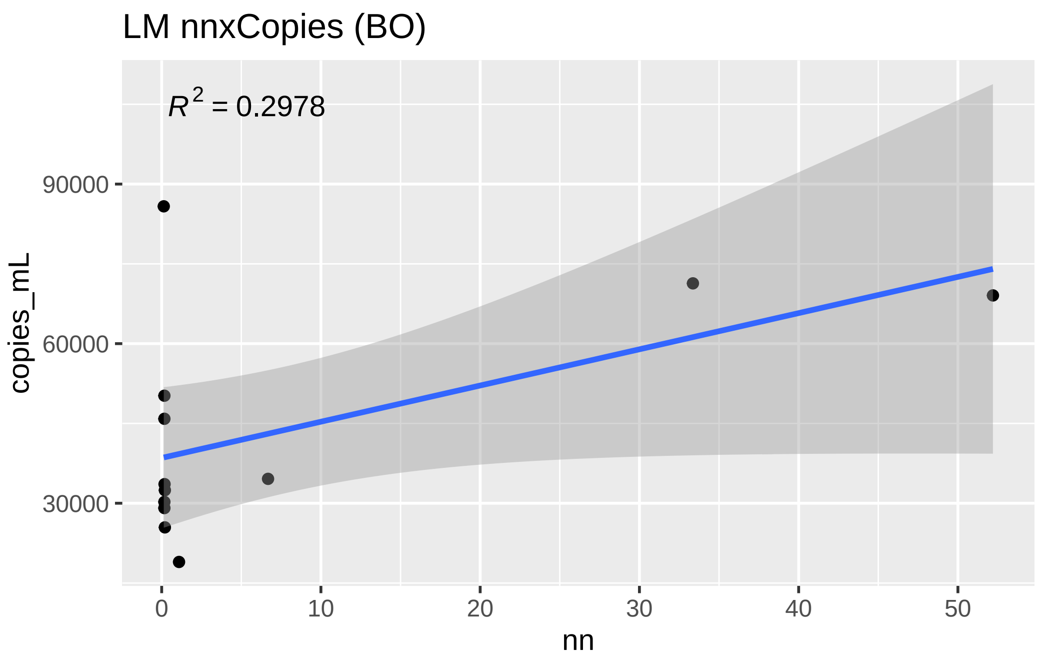
```
`geom_smooth()` using formula = 'y ~ x'
```



NN

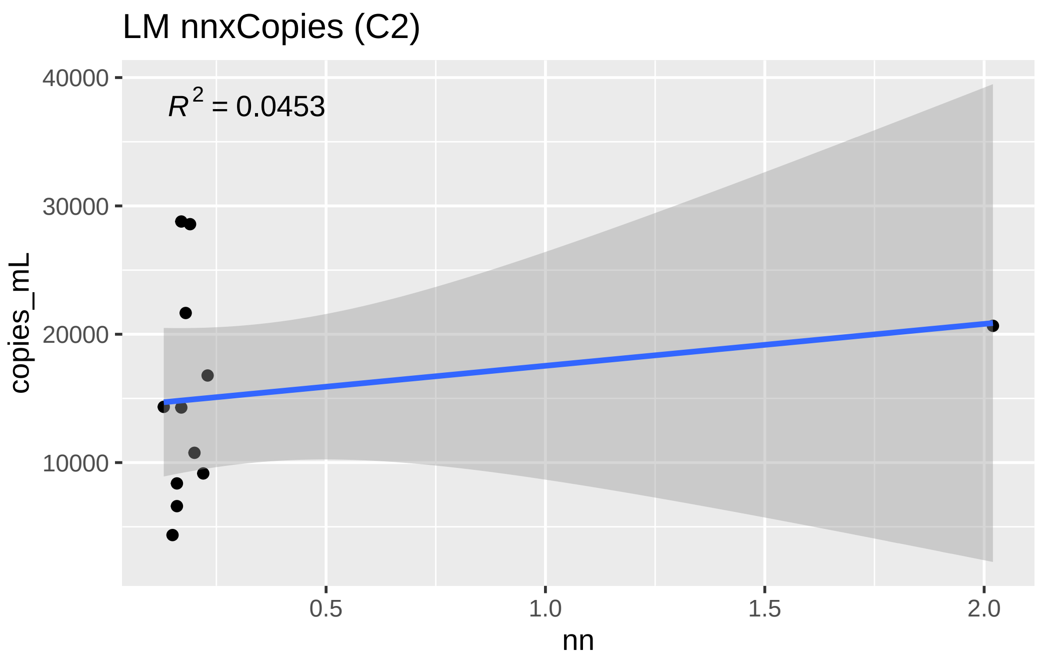
```
b %>% ggplot(aes(x = nn, y = copies_mL)) +
  geom_point() +
  stat_smooth(method = "lm")+
  stat_poly_eq(aes(x = nn, y = copies_mL),
               rr.digits = 4) +
  labs(title = "LM nnxCopies (BO)")
```

```
`geom_smooth()` using formula = 'y ~ x'
```



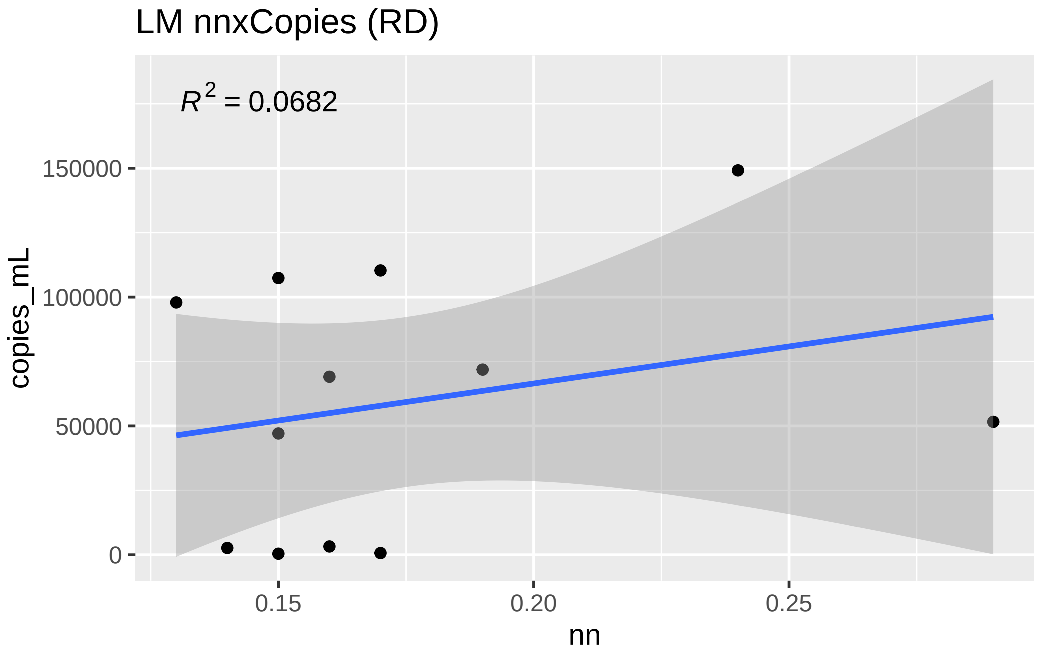
```
c %>% ggplot(aes(x = nn, y = copies_mL)) +
  geom_point() +
  stat_smooth(method = "lm")+
  stat_poly_eq(aes(x = nn, y = copies_mL),
               rr.digits = 4) +
  labs(title = "LM nnxCopies (C2)")
```

```
`geom_smooth()` using formula = 'y ~ x'
```



```
r %>% ggplot(aes(x = nn, y = copies_mL)) +
  geom_point() +
  stat_smooth(method = "lm")+
  stat_poly_eq(aes(x = nn, y = copies_mL),
               rr.digits = 4) +
  labs(title = "LM nnxCopies (RD)")
```

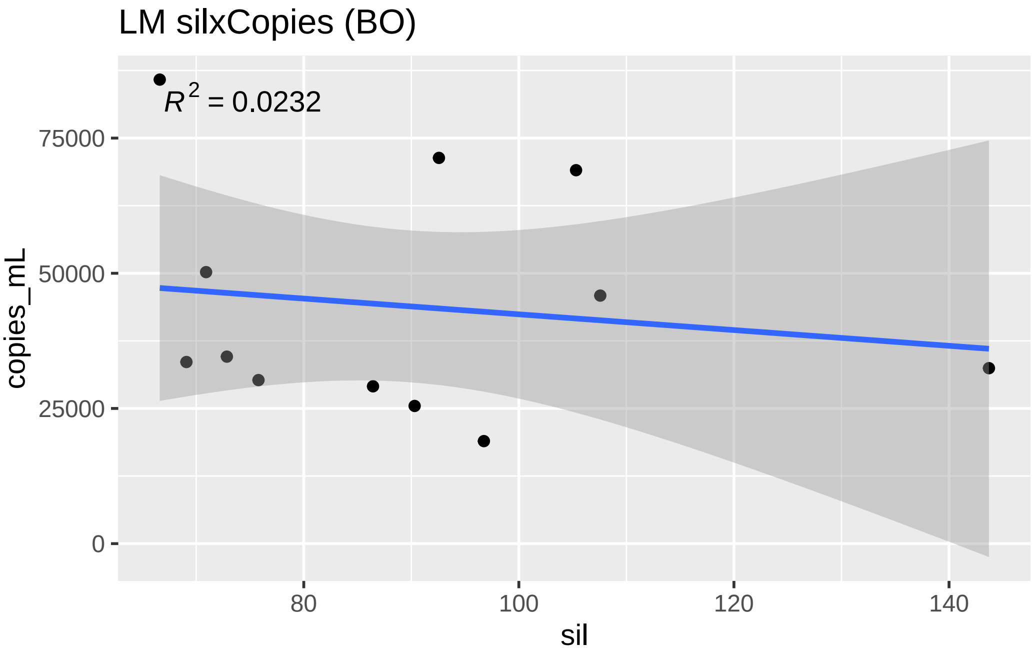
```
`geom_smooth()` using formula = 'y ~ x'
```



Silicate

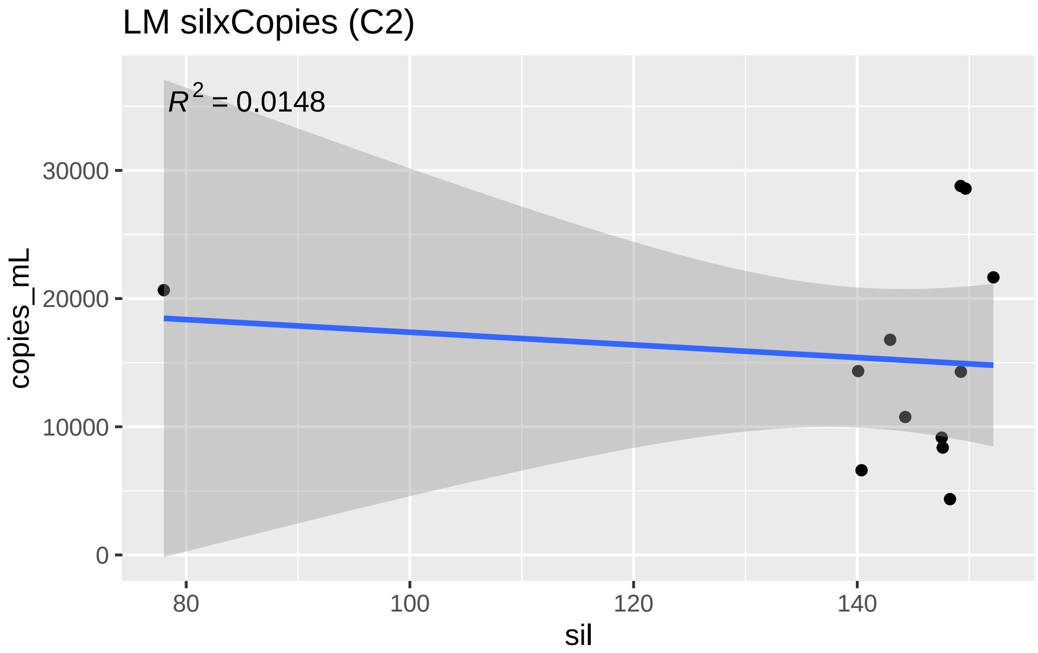
```
b %>% ggplot(aes(x = sil, y = copies_mL)) +
  geom_point() +
  stat_smooth(method = "lm")+
  stat_poly_eq(aes(x = sil, y = copies_mL),
               rr.digits = 4) +
  labs(title = "LM silxCopies (BO)")
```

```
`geom_smooth()` using formula = 'y ~ x'
```



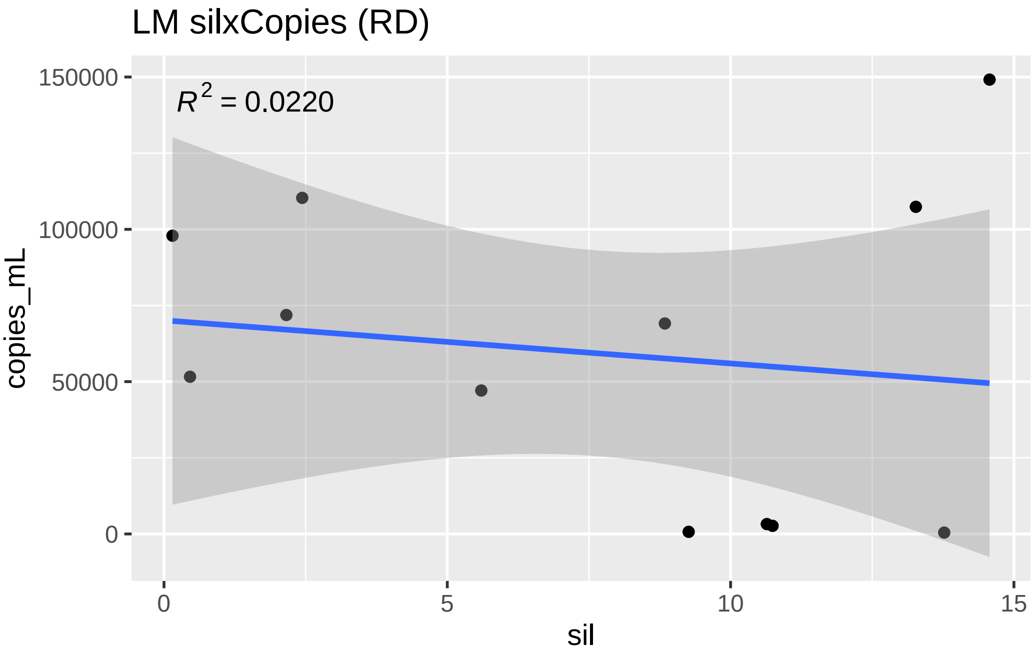
```
c %>% ggplot(aes(x = sil, y = copies_mL)) +
  geom_point() +
  stat_smooth(method = "lm")+
  stat_poly_eq(aes(x = sil, y = copies_mL),
               rr.digits = 4) +
  labs(title = "LM silxCopies (C2)")
```

```
`geom_smooth()` using formula = 'y ~ x'
```



```
r %>% ggplot(aes(x = sil, y = copies_mL)) +
  geom_point() +
  stat_smooth(method = "lm")+
  stat_poly_eq(aes(x = sil, y = copies_mL),
               rr.digits = 4) +
  labs(title = "LM silxCopies (RD)")
```

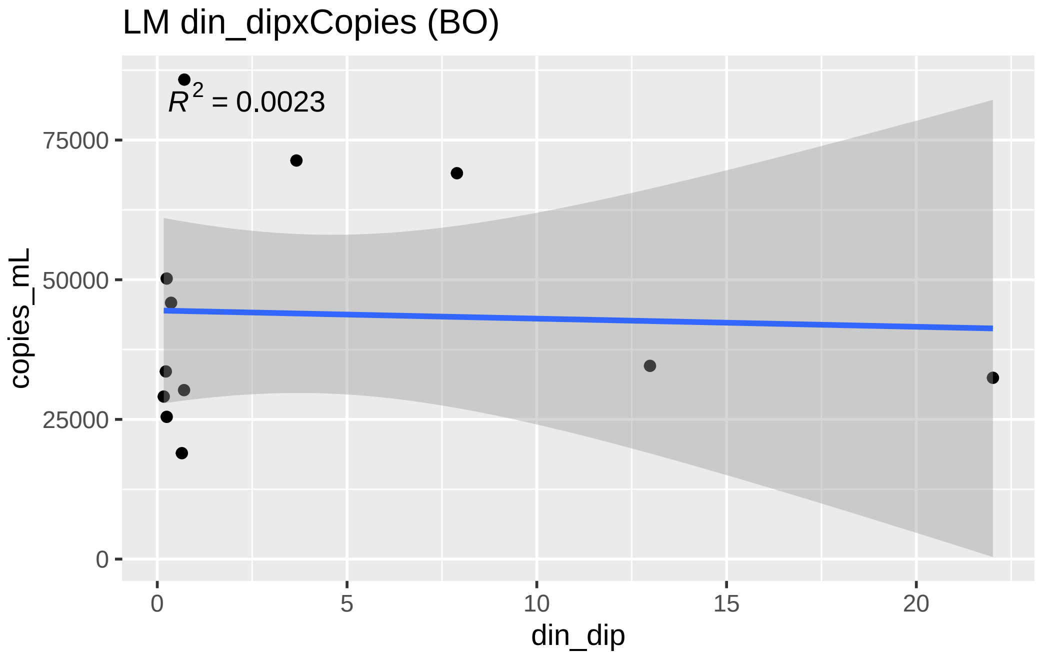
```
`geom_smooth()` using formula = 'y ~ x'
```



DIN:DIP

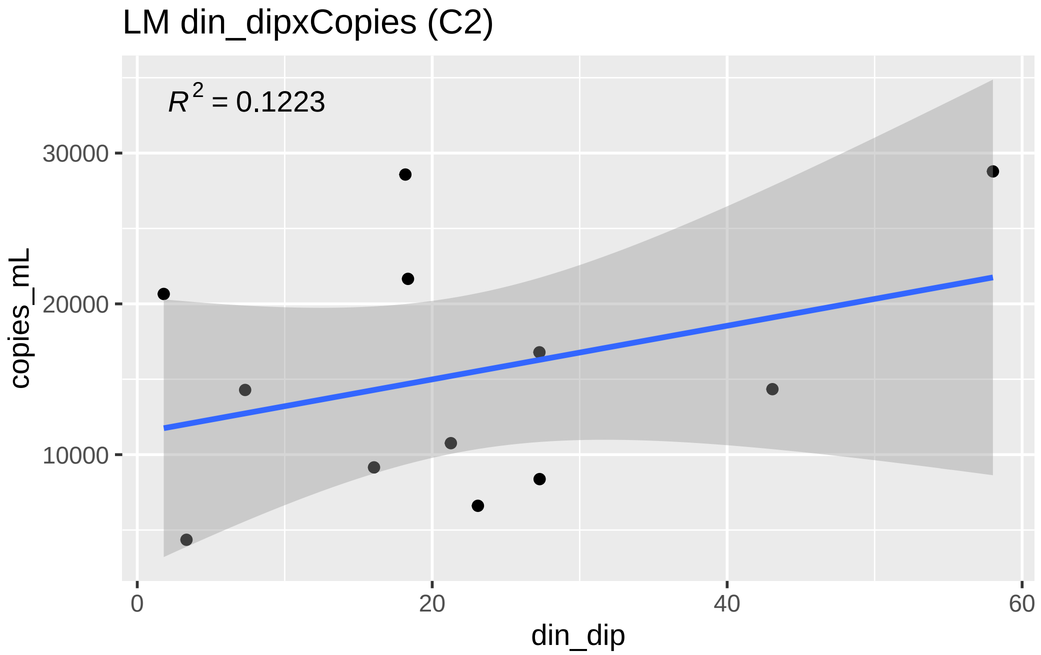
```
b %>% ggplot(aes(x = din_dip, y = copies_mL)) +
  geom_point() +
  stat_smooth(method = "lm")+
  stat_poly_eq(aes(x = din_dip, y = copies_mL),
               rr.digits = 4) +
  labs(title = "LM din_dipxCopies (BO)")
```

```
`geom_smooth()` using formula = 'y ~ x'
```



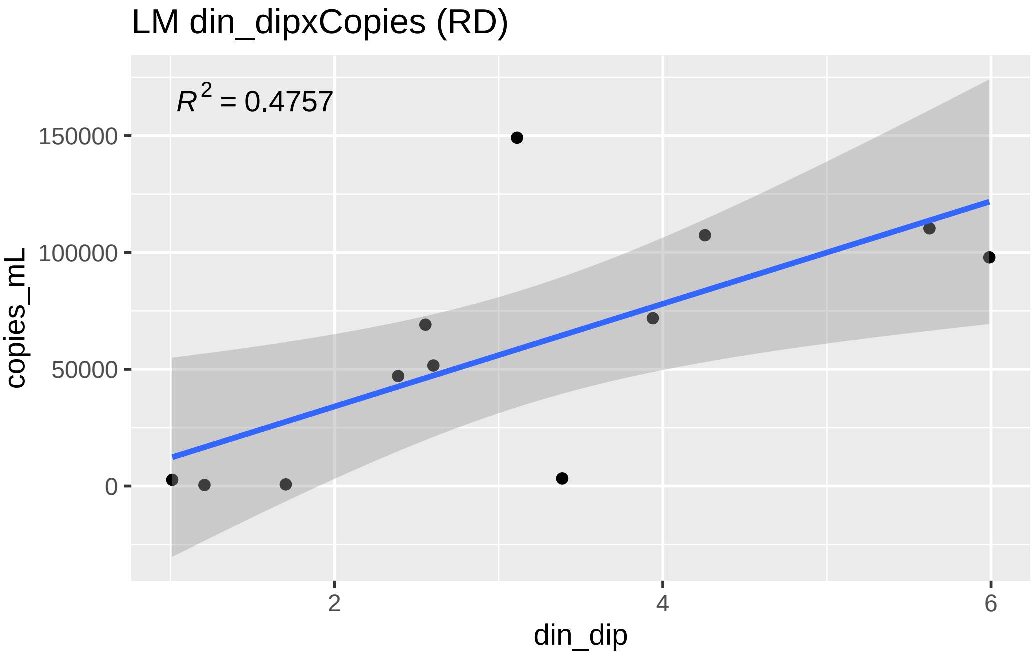
```
c %>% ggplot(aes(x = din_dip, y = copies_mL)) +
  geom_point() +
  stat_smooth(method = "lm")+
  stat_poly_eq(aes(x = din_dip, y = copies_mL),
               rr.digits = 4) +
  labs(title = "LM din_dipxCopies (C2)")
```

```
`geom_smooth()` using formula = 'y ~ x'
```



```
r %>% ggplot(aes(x = din_dip, y = copies_mL)) +
  geom_point() +
  stat_smooth(method = "lm")+
  stat_poly_eq(aes(x = din_dip, y = copies_mL),
               rr.digits = 4) +
  labs(title = "LM din_dipxCopies (RD)")
```

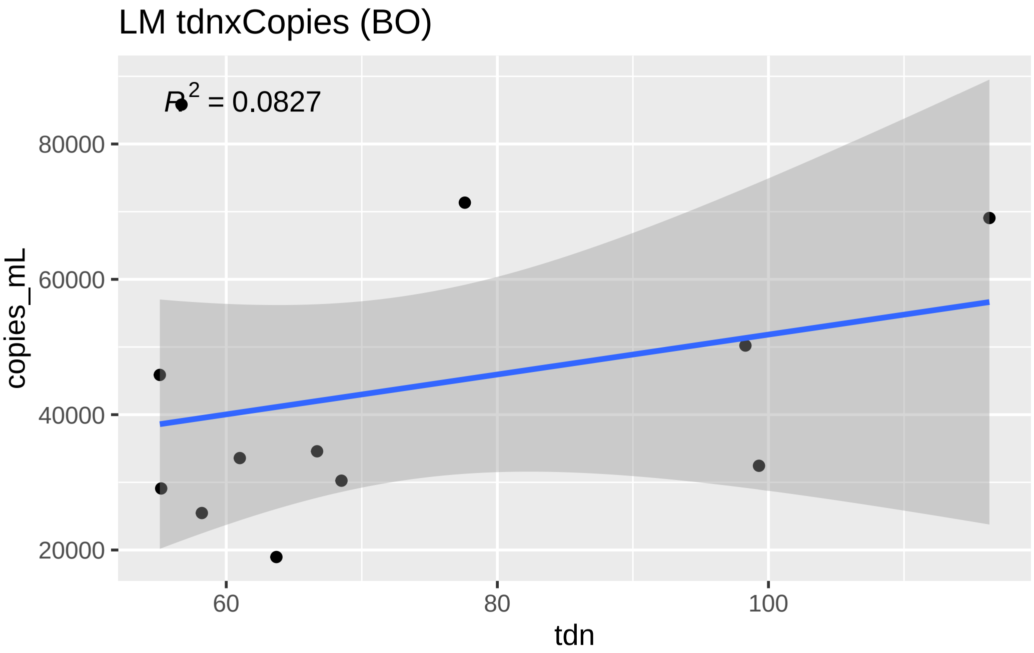
```
`geom_smooth()` using formula = 'y ~ x'
```



TDN

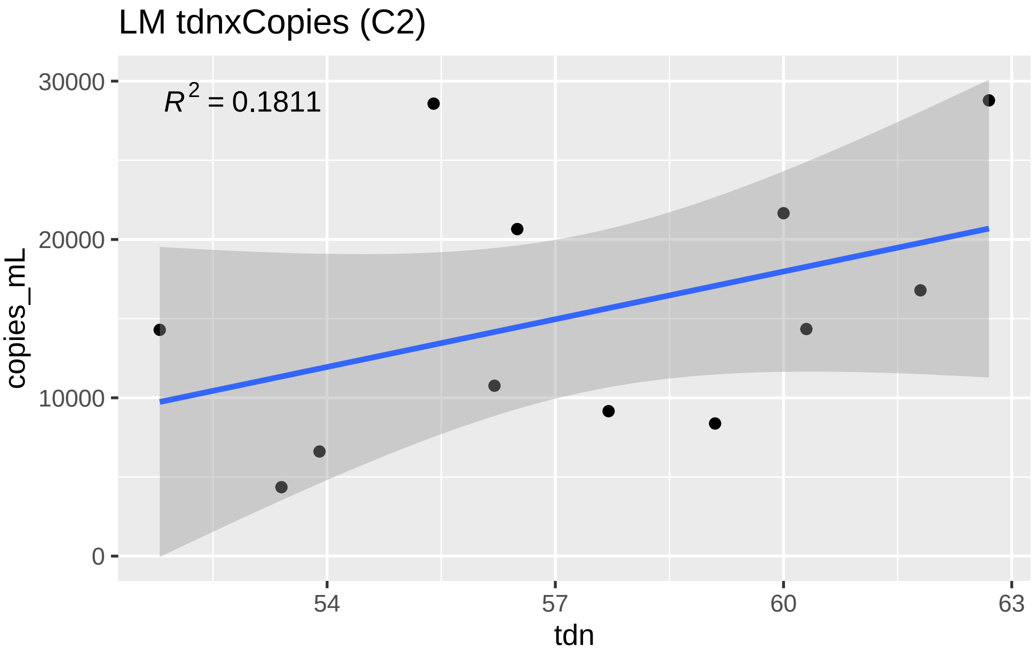
```
b %>% ggplot(aes(x = tdn, y = copies_mL)) +
  geom_point() +
  stat_smooth(method = "lm")+
  stat_poly_eq(aes(x = tdn, y = copies_mL),
               rr.digits = 4) +
  labs(title = "LM tdnxCopies (BO)")
```

```
`geom_smooth()` using formula = 'y ~ x'
```



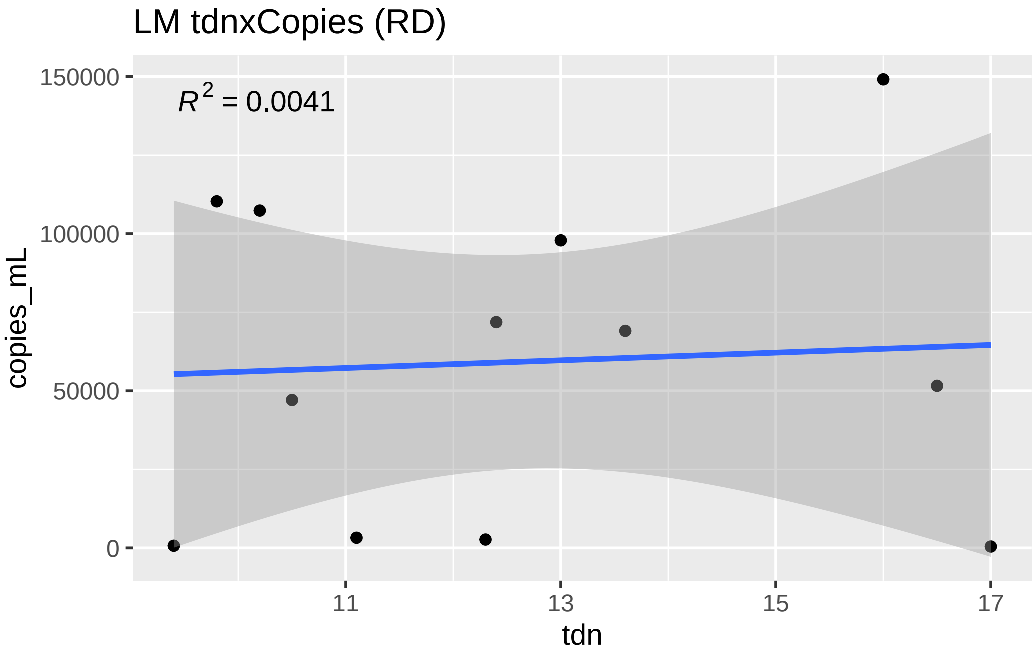
```
c %>% ggplot(aes(x = tdn, y = copies_mL)) +
  geom_point() +
  stat_smooth(method = "lm")+
  stat_poly_eq(aes(x = tdn, y = copies_mL),
               rr.digits = 4) +
  labs(title = "LM tdnxCopies (C2)")
```

```
`geom_smooth()` using formula = 'y ~ x'
```



```
r %>% ggplot(aes(x = tdn, y = copies_mL)) +
  geom_point() +
  stat_smooth(method = "lm")+
  stat_poly_eq(aes(x = tdn, y = copies_mL),
               rr.digits = 4) +
  labs(title = "LM tdnxCopies (RD)")
```

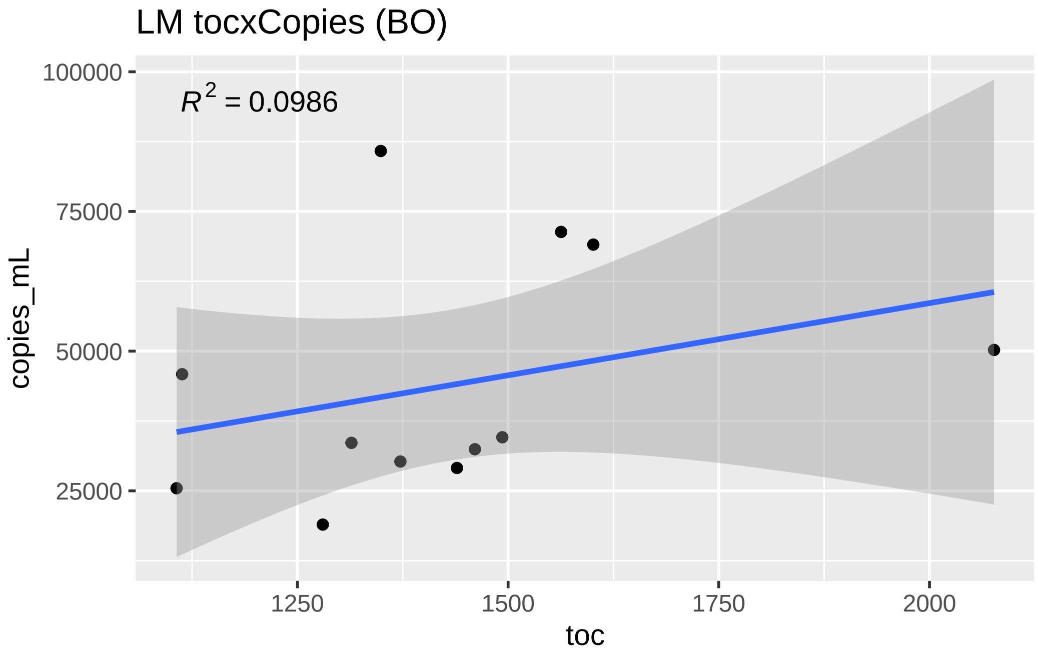
```
`geom_smooth()` using formula = 'y ~ x'
```



TOC

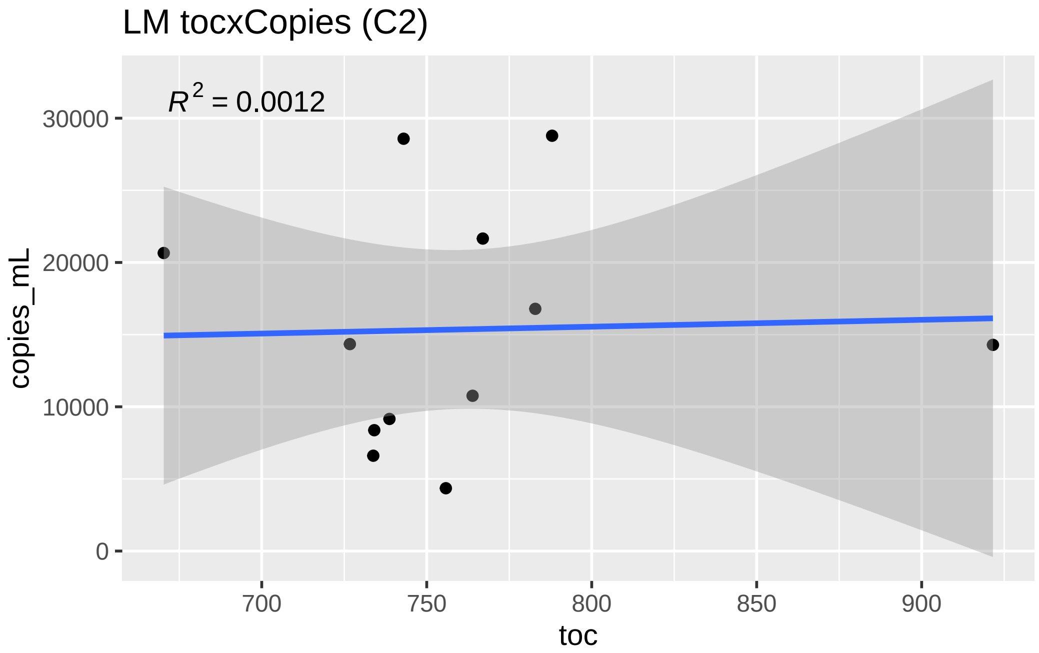
```
b %>% ggplot(aes(x = toc, y = copies_mL)) +
  geom_point() +
  stat_smooth(method = "lm")+
  stat_poly_eq(aes(x = toc, y = copies_mL),
               rr.digits = 4) +
  labs(title = "LM tocxCopies (BO)")
```

```
`geom_smooth()` using formula = 'y ~ x'
```



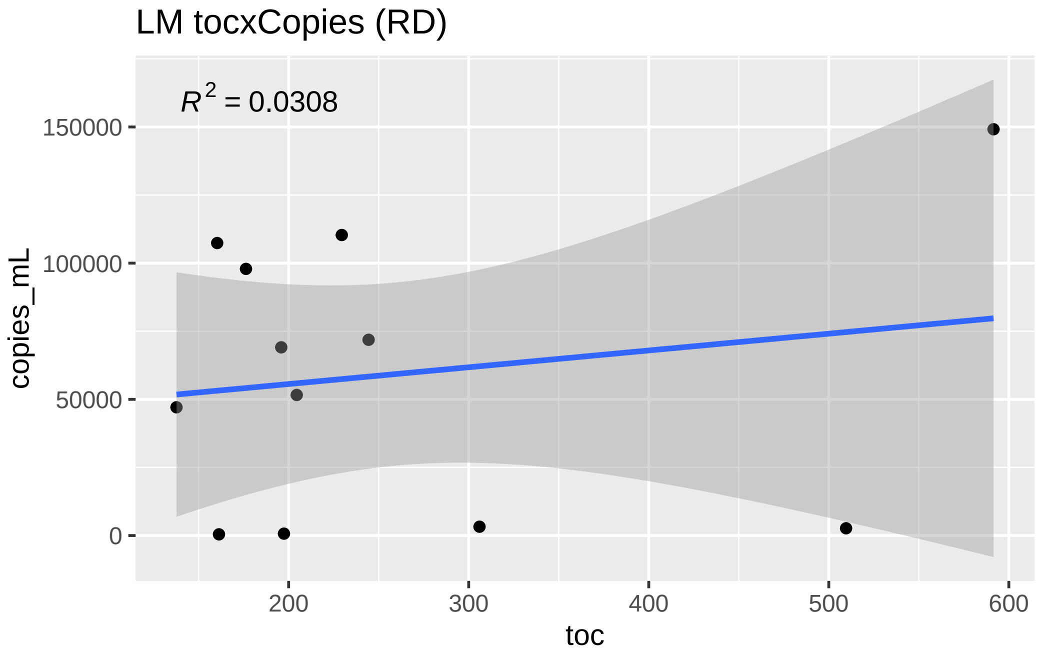
```
c %>% ggplot(aes(x = toc, y = copies_mL)) +
  geom_point() +
  stat_smooth(method = "lm")+
  stat_poly_eq(aes(x = toc, y = copies_mL),
               rr.digits = 4) +
  labs(title = "LM tocxCopies (C2)")
```

```
`geom_smooth()` using formula = 'y ~ x'
```



```
r %>% ggplot(aes(x = toc, y = copies_mL)) +
  geom_point() +
  stat_smooth(method = "lm")+
  stat_poly_eq(aes(x = toc, y = copies_mL),
               rr.digits = 4) +
  labs(title = "LM tocxCopies (RD)")
```

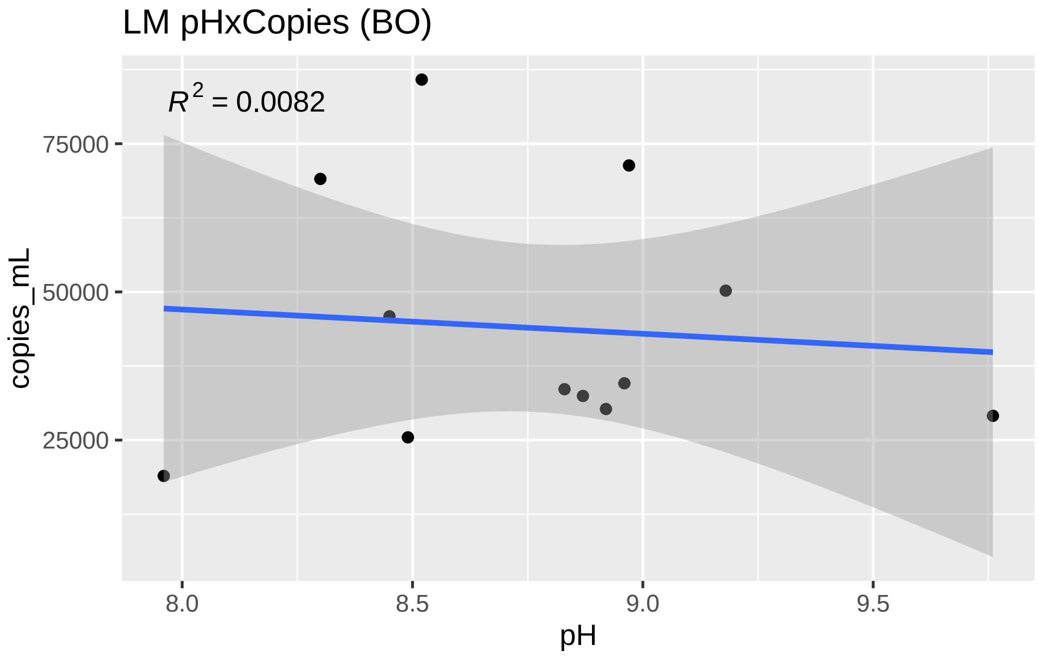
```
`geom_smooth()` using formula = 'y ~ x'
```



pH

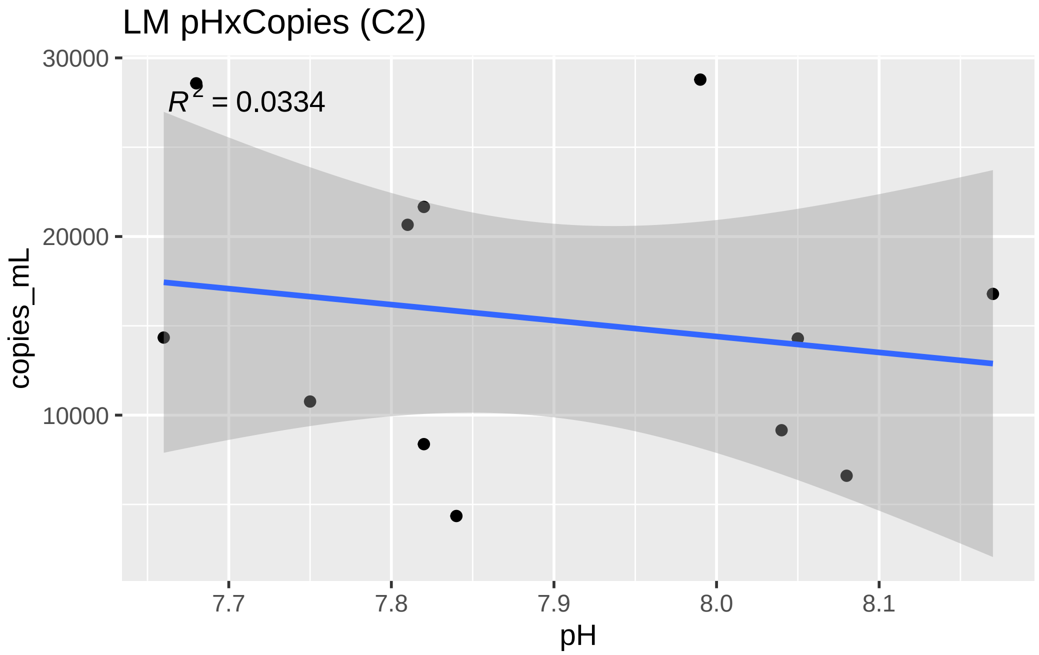
```
b %>% ggplot(aes(x = pH, y = copies_mL)) +
  geom_point() +
  stat_smooth(method = "lm")+
  stat_poly_eq(aes(x = pH, y = copies_mL),
               rr.digits = 4) +
  labs(title = "LM pHxCopies (BO)")
```

```
`geom_smooth()` using formula = 'y ~ x'
```



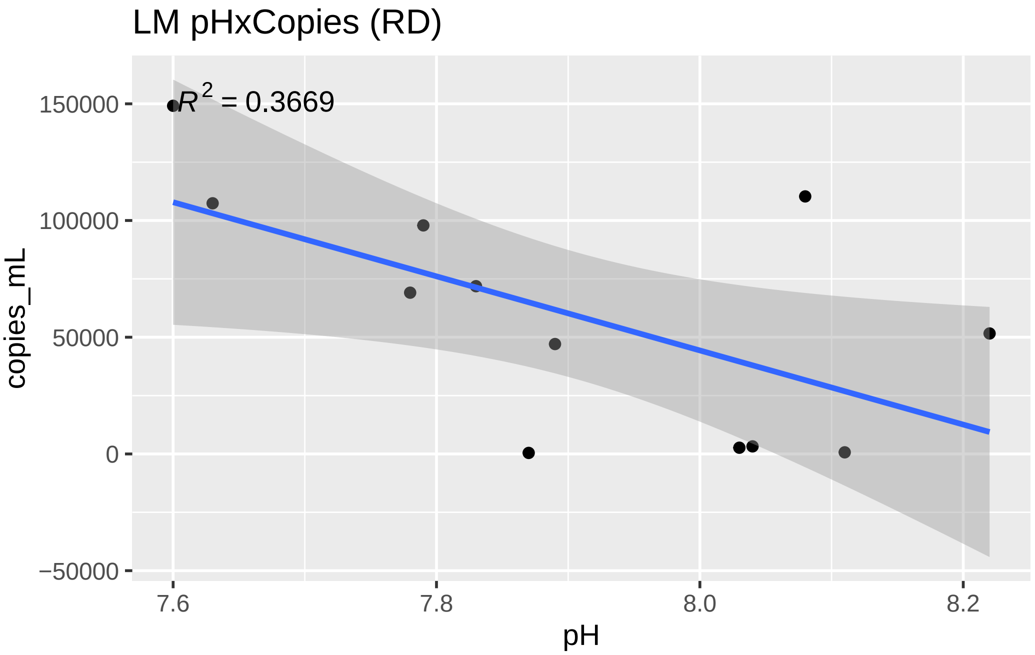
```
c %>% ggplot(aes(x = pH, y = copies_mL)) +
  geom_point() +
  stat_smooth(method = "lm")+
  stat_poly_eq(aes(x = pH, y = copies_mL),
               rr.digits = 4) +
  labs(title = "LM pHxCopies (C2)")
```

```
`geom_smooth()` using formula = 'y ~ x'
```



```
r %>% ggplot(aes(x = pH, y = copies_mL)) +
  geom_point() +
  stat_smooth(method = "lm")+
  stat_poly_eq(aes(x = pH, y = copies_mL),
               rr.digits = 4) +
  labs(title = "LM pHxCopies (RD)")
```

```
`geom_smooth()` using formula = 'y ~ x'
```


LM pHxCopies (RD)
$R^2 = 0.3669$
150000
100000
50000
0
−50000
copies_mL
7.6
7.8
8.0
8.2
pH