

512 Project Part I

Jared Adam

Due Sept 27

The number of conflicts with dplyr from all of the packages we must download is becoming annoying and breaking code.

Part I (512 only, project proposal, 25 pts):

- 1) Read in your data set and run dim on it:

```
set.seed(654321)
s21 <- read_csv('data/2021 Sentinel Prey Assessment.csv')
s22 <- read_csv("data/PSA_CE2_SentinelPrey.csv")
s23 <- read_csv('data/PSA_Sent.prey.2023.csv')
```

I need to get total predation into a column as a binary. 1 = predation, 0 = not

2021 cleaning

```
s21
```

A tibble: 5,281 × 17

	location	date	n.weather	d.weather	growth_stage	plot_id	rep.block
treatment							
##	<chr>	<chr>	<dbl>	<chr>	<chr>	<dbl>	<dbl>
<dbl>							
##	1 PA	6/16/2...	15.3	18.44	V3	101	1
1							
##	2 PA	6/16/2...	15.3	18.44	V3	101	1
1							
##	3 PA	6/16/2...	15.3	18.44	V3	101	1
1							
##	4 PA	6/16/2...	15.3	18.44	V3	101	1
1							
##	5 PA	6/16/2...	15.3	18.44	V3	101	1
1							
##	6 PA	6/16/2...	15.3	18.44	V3	101	1
1							
##	7 PA	6/16/2...	15.3	18.44	V3	102	1
3							
##	8 PA	6/16/2...	15.3	18.44	V3	102	1
3							
##	9 PA	6/16/2...	15.3	18.44	V3	102	1
3							
##	10 PA	6/16/2...	15.3	18.44	V3	102	1

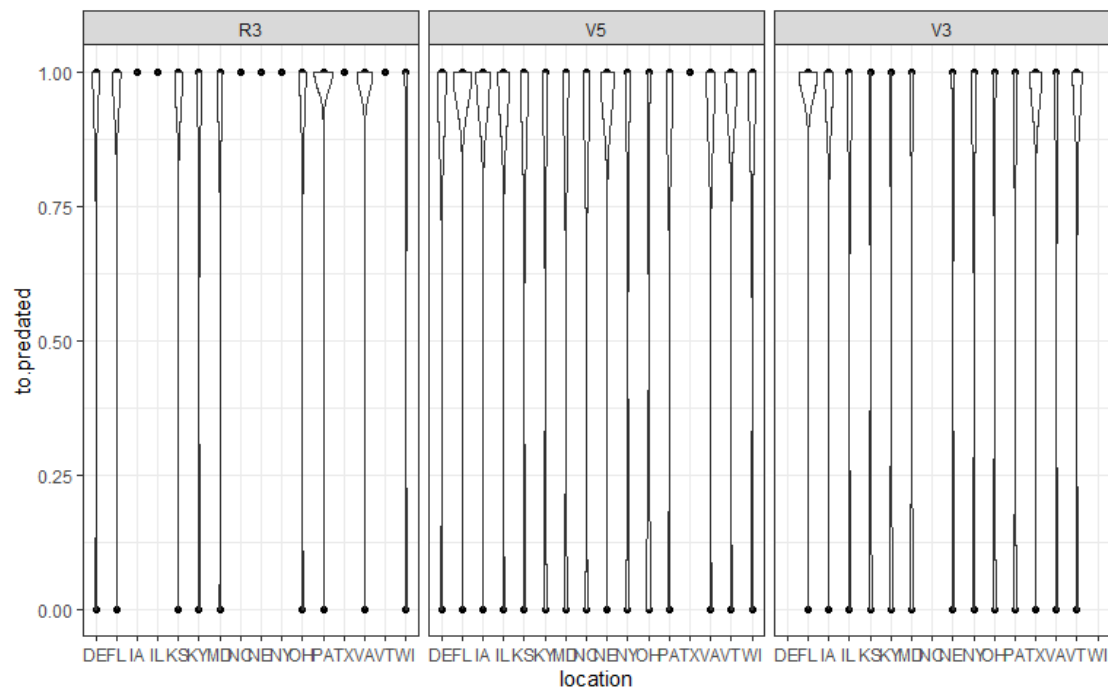
```

3
## # i 5,271 more rows
## # i 9 more variables: row <dbl>, sample <dbl>, n.absent <chr>, n.partial
<chr>,
## #   n.predated <chr>, d.absent <chr>, d.partial <chr>, d.predated <chr>,
## #   to.predated <chr>

clean21 <- s21 %>%
  mutate(year = '2021') %>%
  dplyr::select(location, year, growth_stage, plot_id, rep.block, treatment,
to.predated) %>%
  mutate(to.predated = as.double(to.predated)) %>%
  dplyr::rename(block = rep.block) %>%
  group_by(location, year, growth_stage, plot_id, block, treatment) %>%
  # dplyr::summarise(total = sum(to.predated)) %>%
  na.omit() %>%
  mutate(treatment = case_when(
    treatment == '33' ~ '3',
    .default = as.factor(treatment))) %>%
  dplyr::filter(treatment != '6',
    treatment != '7',
    treatment != '8') %>%
  mutate_at(vars(1:6), as.factor)

ggplot(clean21, aes(x = location, y = to.predated))+
  geom_point()+
  geom_violin()+
  facet_wrap(~growth_stage)

```



```
# 2022 cleaning
```

```
s22
```

```
## # A tibble: 3,246 × 19
```

```
##   location date      am.weather pm.weather growth_stage plotid block  
treatment
```

```
##   <chr>      <chr>          <dbl> <chr>      <chr>          <dbl> <dbl>  
<dbl>
```

```
## 1 PA        6/22/2022      22.7 26.1      V3          101      1
```

```
1
```

```
## 2 PA        6/22/2022      22.7 26.1      V3          101      1
```

```
1
```

```
## 3 PA        6/22/2022      22.7 26.1      V3          101      1
```

```
1
```

```
## 4 PA        6/22/2022      22.7 26.1      V3          101      1
```

```
1
```

```
## 5 PA        6/22/2022      22.7 26.1      V3          101      1
```

```
1
```

```
## 6 PA        6/22/2022      22.7 26.1      V3          101      1
```

```
1
```

```
## 7 PA        6/22/2022      22.7 26.1      V3          102      1
```

```
3
```

```
## 8 PA        6/22/2022      22.7 26.1      V3          102      1
```

```
3
```

```
## 9 PA        6/22/2022      22.7 26.1      V3          102      1
```

```
3
```

```
## 10 PA       6/22/2022      22.7 26.1      V3          102      1
```

```
3
```

```
## # i 3,236 more rows
```

```
## # i 11 more variables: row <dbl>, sample <dbl>, am.absent <chr>,
```

```
## #   am.partial <chr>, am.predators <chr>, pm.absent <chr>, pm.partial  
<chr>,
```

```
## #   pm.predators <chr>, to.predated <dbl>, n.predated <dbl>, d.predated  
<dbl>
```

```
unique(s22$treatment)
```

```
## [1] 1 3 2 4
```

```
unique(s22$growth_stage)
```

```
## [1] "V3" "V5" "R3" "R2"
```

```
clean22 <- s22 %>%
```

```
  mutate(year = '2022') %>%
```

```
  dplyr::select(location, year, growth_stage, plotid, block, treatment,  
to.predated) %>%
```

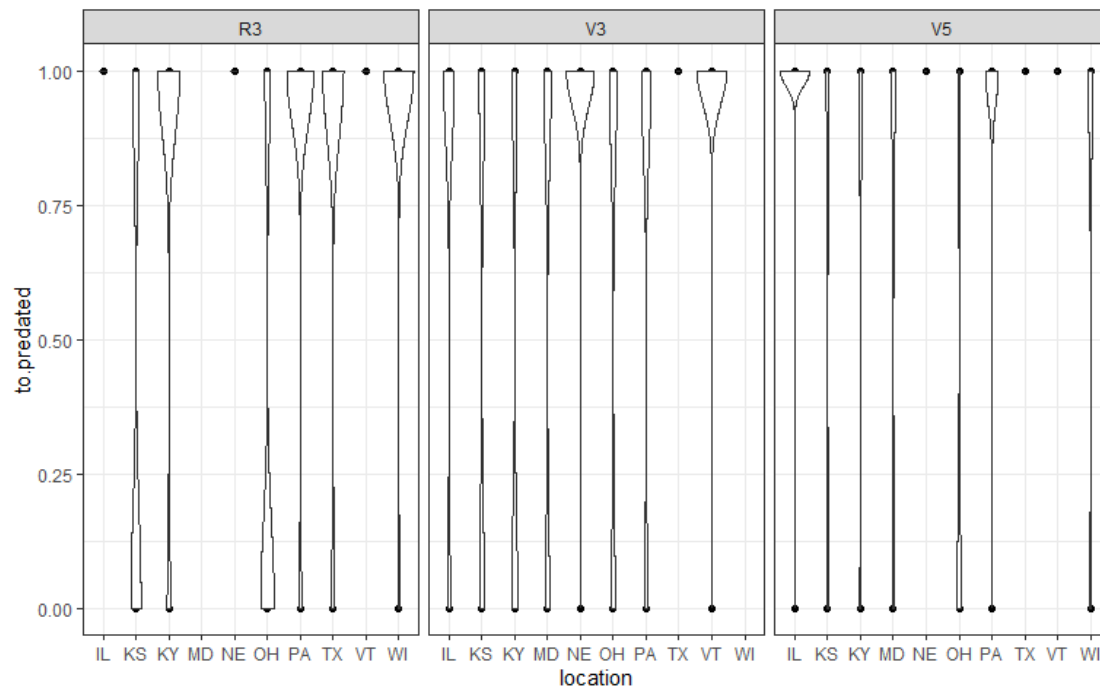
```
  dplyr::rename(plot_id = plotid) %>%
```

```
  mutate(growth_stage = case_when(growth_stage == 'R2' ~ 'R3',  
                                .default = as.character(growth_stage))) %>%
```

```
  dplyr::group_by(location, year, growth_stage, plot_id, block, treatment)
```

```
%>%
  # dplyr::summarise(total = sum(to.predated)) %>%
  mutate_at(vars(1:6), as.factor)

ggplot(clean22, aes(x = location, y = to.predated))+
  geom_point()+
  geom_violin()+
  facet_wrap(~growth_stage)
```

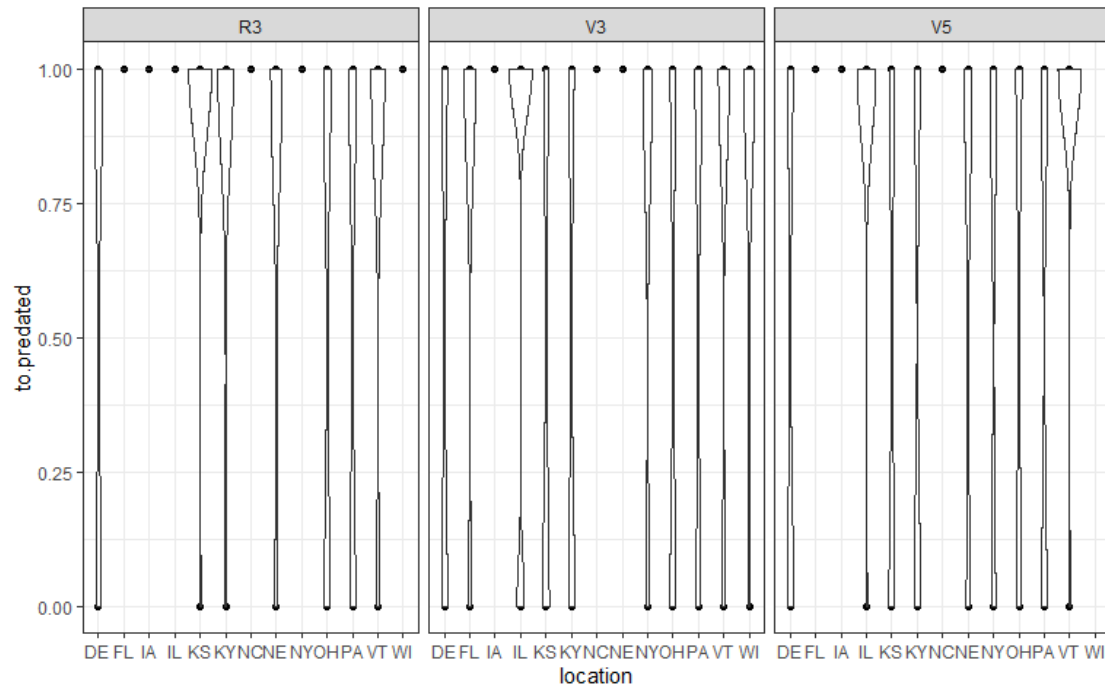


```
# 2023 cleaning

clean23 <- s23 %>%
  mutate(year = '2023') %>%
  relocate(am.partial, am.absent, pm.partial, pm.absent) %>%
  mutate_at(vars(1:4), as.double) %>%
  mutate(to.predated = if_else(am.partial | am.absent | pm.partial |
pm.absent == 1, 1, 0)) %>%
  relocate(to.predated) %>%
  mutate(growth_stage = case_when((location == 'NC' & date == '7/20/2023') ~
'R3',
                                .default = as.character(growth_stage))) %>%
  dplyr::select(location, year, growth_stage, plotid, block, treatmetn,
to.predated) %>%
  dplyr::rename(plot_id = plotid,
                treatment = treatmetn) %>%
  distinct() %>%
  group_by(location, year, growth_stage, plot_id, block, treatment) %>%
  na.omit() %>%
  filter(treatment != 5) %>%
```

```
mutate_at(vars(1:6), as.factor)
```

```
ggplot(clean23, aes(x = location, y = to.predated)) +  
  geom_point() +  
  geom_violin() +  
  facet_wrap(~growth_stage)
```



and in the darkness, bind them

```
sent <- rbind(clean21, clean22, clean23)  
as_tibble(sent)
```

```
## # A tibble: 9,227 × 7
```

```
##   location year growth_stage plot_id block treatment to.predated  
##   <fct>    <fct> <fct>      <fct>  <fct> <fct>          <dbl>  
## 1 PA      2021 V3        101     1     1            0  
## 2 PA      2021 V3        101     1     1            0  
## 3 PA      2021 V3        101     1     1            0  
## 4 PA      2021 V3        101     1     1            0  
## 5 PA      2021 V3        101     1     1            0  
## 6 PA      2021 V3        101     1     1            0  
## 7 PA      2021 V3        102     1     3            1  
## 8 PA      2021 V3        102     1     3            1  
## 9 PA      2021 V3        102     1     3            0  
## 10 PA     2021 V3        102     1     3            1
```

```
## # i 9,217 more rows
```

```
dim(sent)
```

```
## [1] 9227    7
```

- 2) Prepare a short description of your data set (source if published paper exists), especially providing the study design, sample size, and variables of primary interest. If there is random sampling, note the population sampled from. If there is random assignment, note how and for which variable(s).

Data: Sentinel Prey assessment of arthropod-predator activity in corn fields.

These data come from the Precision Sustainable Agriculture effort through the USDA. I am the lead on the entomology component of this project and responsible for analyzing this three year data set which spans multiple states. This effort began during my Master's degree, but I only analyzed Pennsylvania data for my thesis.

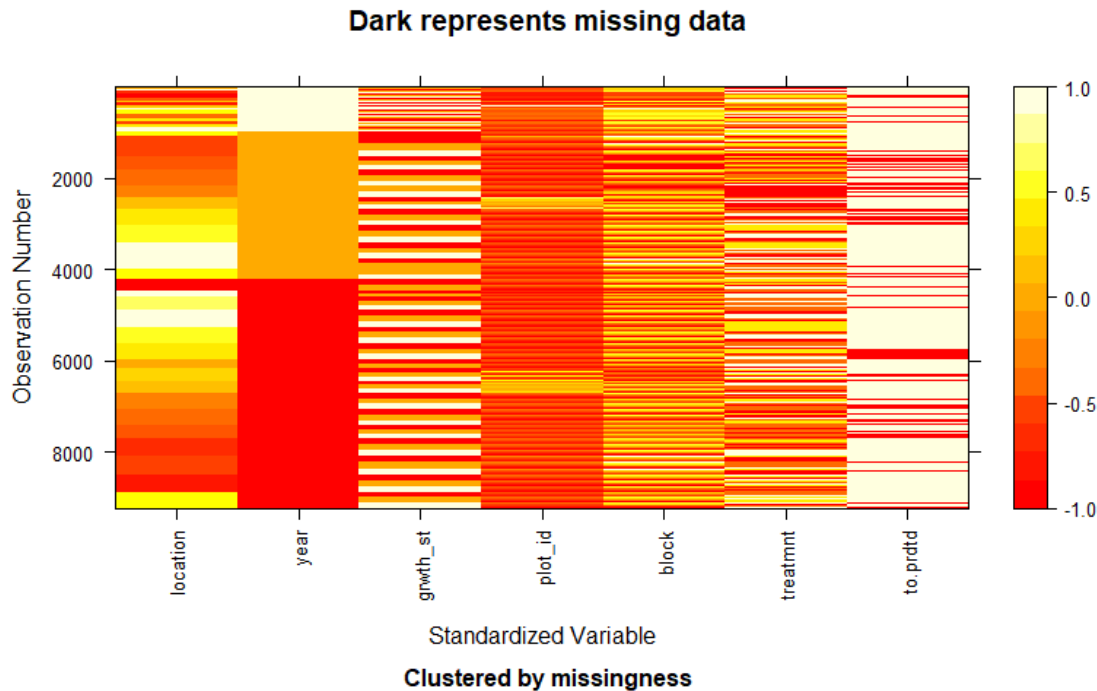
Study design: **Treatments** = 4; No cover crop, early-terminated cover crop, late-terminated cover crop, planting green **Plots** = 20; 5 blocks composed of 4 plots each = 20 plots / study site / year **Years** = 3 (2021,2022,2023) **Locations** = This project comprises 16 states. Not all states collected sentinel prey data every year. **Effort** = Data were collected at three corn growth stages / year (V3,V5,R3). **Sample** = 6 sentinel prey traps were placed in each plot = 120 samples collected / growth stage. Total sample effort per state per season = 360 samples.

Variables: **Response** = Total level of predation. This is transformed into a proportion over a constant total. Binomial response **Explanatory** = Crop growth stage (timing, three levels) and treatment (four levels). **Random effects** = Block and location? I am unsure how to use location. I am not interested in seeing how each state differs in the analysis because that is *not* a research question. I plan to go through and conduct each state's own analysis later. **Repeated measure** = I visit the same trap three times throughout the year. This feels like a term I should identify. If I were to choose, I would select *growth stage*.

Plots were randomly assigned to each block. Field sites were as random as they could be at each respective research station. Sentinel prey traps were placed between pre-determined rows and at specific length intervals within each plot to maintain consistency.

- 3) Make a `missing_data.frame` plot of your data set and explain any missing values indicated:

```
library(mi)
# make an object of the missing df and then present the image
tdf <- missing_data.frame(data.frame(sent))
image(tdf)
```

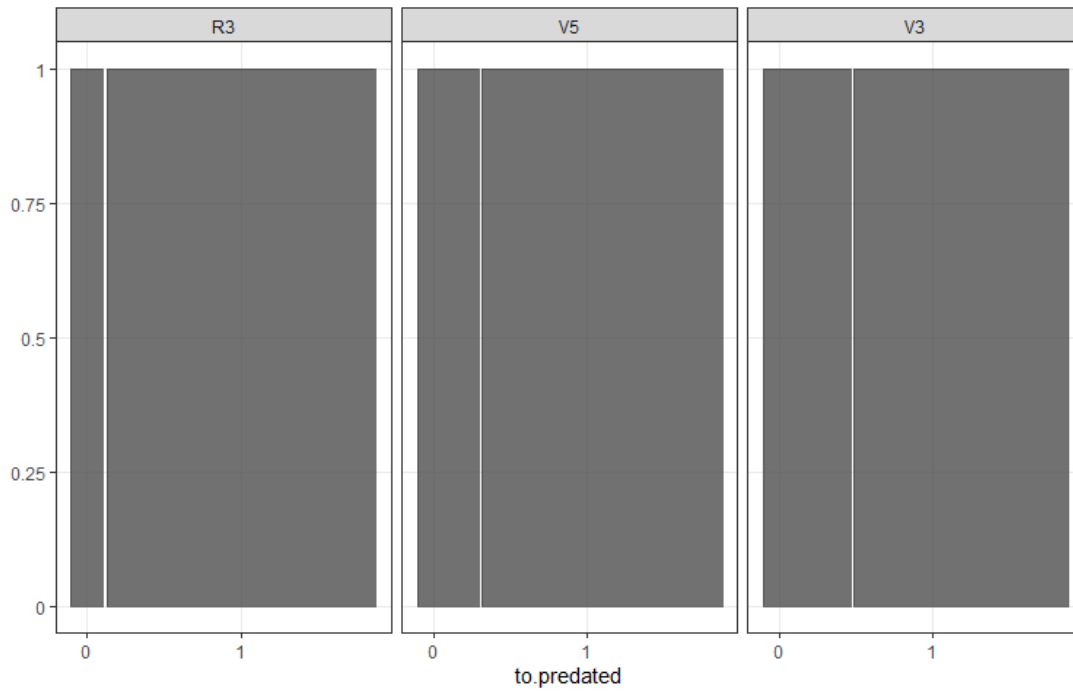


- 4) Discuss any other use in classes or theses for the data - either that you have used it for or are currently working on for future submissions.

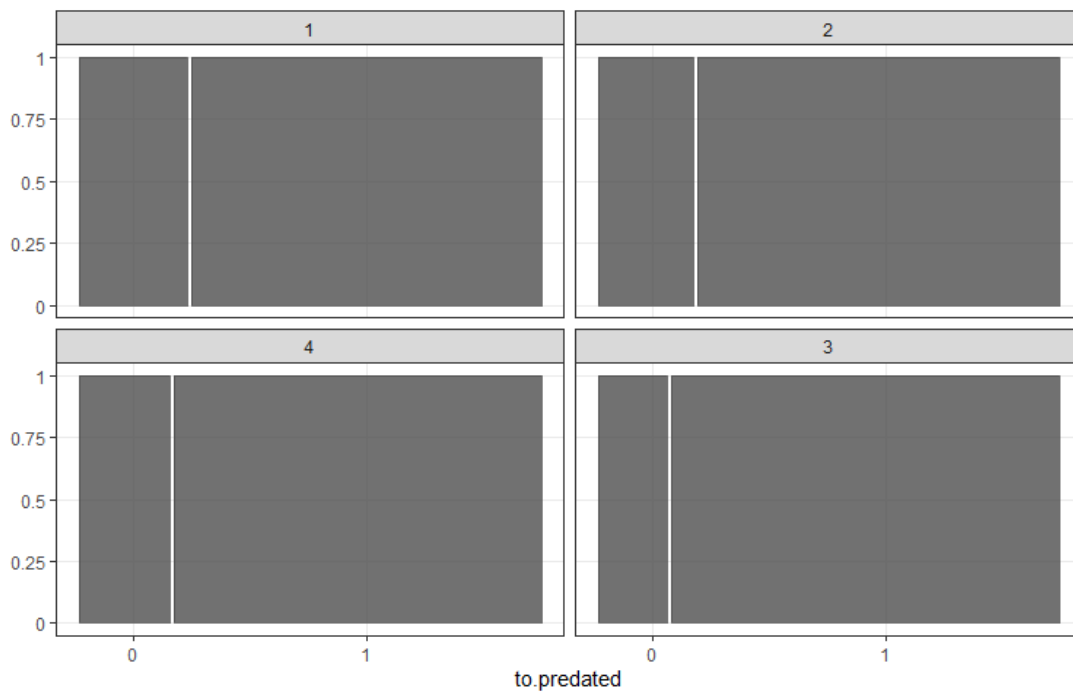
I am working on this for a publication. There is no published paper yet. None of the code from that is used here. This analysis is for all of the states combined, but in the future, I plan to run each state individually with their three years of data. I suspect results to differ based on some regional grouping (e.g., growing degree days, growth region, etc.), but am yet to decide what I will use. For now, I am mainly interested in the treatment and growth stage effects on the whole data set.

- 5) Provide at least one display of the data, focusing on the response of interest versus a predictor. If you have multiple predictors, try to plot the response versus those too.

```
library(ggmosaic)
sent %>%
  ggplot() +
  geom_mosaic(aes(x = product(to.predated))) +
  facet_wrap(~growth_stage) +
  scale_fill_colorblind()
```



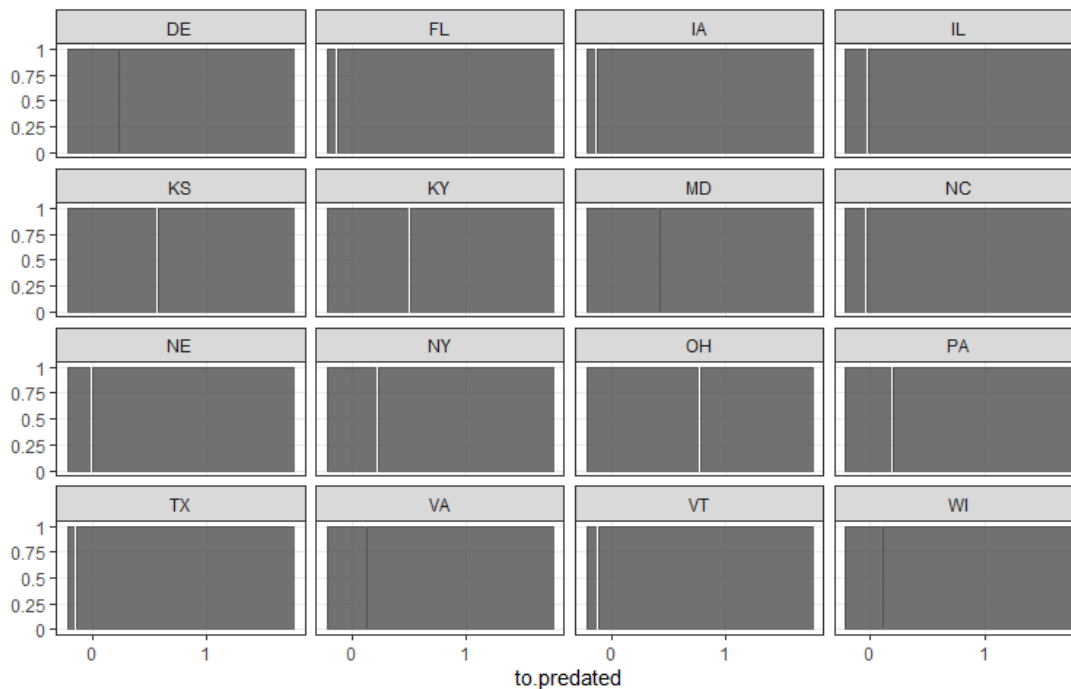
```
sent %>%
  ggplot() +
  geom_mosaic(aes(x = product(to.predated)))+
  facet_wrap(~treatment)+
  scale_fill_colorblind()
```



```
sent %>%
  ggplot() +
```



```
geom_mosaic(aes(x = product(to.predated))) +
facet_wrap(~location) +
scale_fill_colorblind()
```



- 6) Provide an initial model you hope to fit (does not need to be fit). If you fit a model, add a model summary and effects plot.

sent

```
## # A tibble: 9,227 × 7
## # Groups:   location, year, growth_stage, plot_id, block, treatment
## [2,091]
##   location year growth_stage plot_id block treatment to.predated
##   <fct>    <fct> <fct>      <fct>  <fct> <fct>      <dbl>
## 1 PA      2021 V3      101     1     1          0
## 2 PA      2021 V3      101     1     1          0
## 3 PA      2021 V3      101     1     1          0
## 4 PA      2021 V3      101     1     1          0
## 5 PA      2021 V3      101     1     1          0
## 6 PA      2021 V3      101     1     1          0
## 7 PA      2021 V3      102     1     3          1
## 8 PA      2021 V3      102     1     3          1
## 9 PA      2021 V3      102     1     3          0
## 10 PA     2021 V3      102     1     3          1
## # i 9,217 more rows

nr_m1 <- glm(to.predated ~ treatment * growth_stage + location, family =
binomial, data = sent)
summary(nr_m1)$coefficients
```

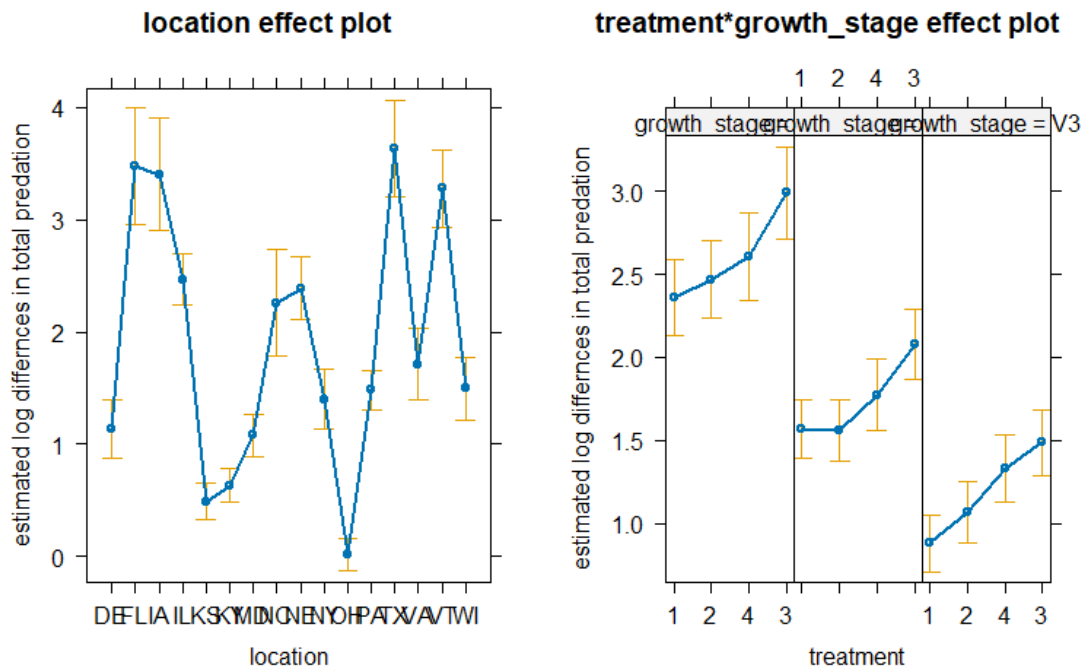
##	Estimate	Std. Error	z value	Pr(> z)
## (Intercept)	1.63753535	0.1698622	9.64037525	5.399025e-22
## treatment2	0.11044854	0.1608715	0.68656392	4.923576e-01
## treatment4	0.24534439	0.1728598	1.41932619	1.558039e-01
## treatment3	0.63175906	0.1784895	3.53947559	4.009228e-04
## growth_stageV5	-0.79041893	0.1403518	-5.63169952	1.784424e-08
## growth_stageV3	-1.48086005	0.1409199	-10.50852583	7.891746e-26
## locationFL	2.35181141	0.2966652	7.92749461	2.236112e-15
## locationIA	2.27166745	0.2895927	7.84435315	4.351908e-15
## locationIL	1.33557301	0.1766371	7.56111315	3.996344e-14
## locationKS	-0.63968220	0.1573252	-4.06598808	4.782938e-05
## locationKY	-0.49632978	0.1533556	-3.23646258	1.210211e-03
## locationMD	-0.05200588	0.1634877	-0.31810264	7.504071e-01
## locationNC	1.12690999	0.2772448	4.06467526	4.809938e-05
## locationNE	1.25781603	0.1943238	6.47278283	9.621419e-11
## locationNY	0.27125927	0.1890008	1.43522844	1.512220e-01
## locationOH	-1.11126167	0.1515812	-7.33113108	2.282183e-13
## locationPA	0.35552901	0.1607051	2.21230749	2.694543e-02
## locationTX	2.50149329	0.2558608	9.77677453	1.416529e-22
## locationVA	0.58197422	0.2091036	2.78318596	5.382794e-03
## locationVT	2.14800102	0.2219601	9.67742106	3.760829e-22
## locationWI	0.36672697	0.1940066	1.89028062	5.872044e-02
## treatment2:growth_stageV5	-0.11619939	0.2034004	-0.57128384	5.678073e-01
## treatment4:growth_stageV5	-0.03870848	0.2183526	-0.17727508	8.592923e-01
## treatment3:growth_stageV5	-0.12008656	0.2226533	-0.53934335	5.896500e-01
## treatment2:growth_stageV3	0.07737351	0.2034816	0.38024812	7.037612e-01
## treatment4:growth_stageV3	0.20765365	0.2165277	0.95901665	3.375504e-01
## treatment3:growth_stageV3	-0.02188705	0.2203358	-0.09933497	9.208723e-01

confint(nr_m1)

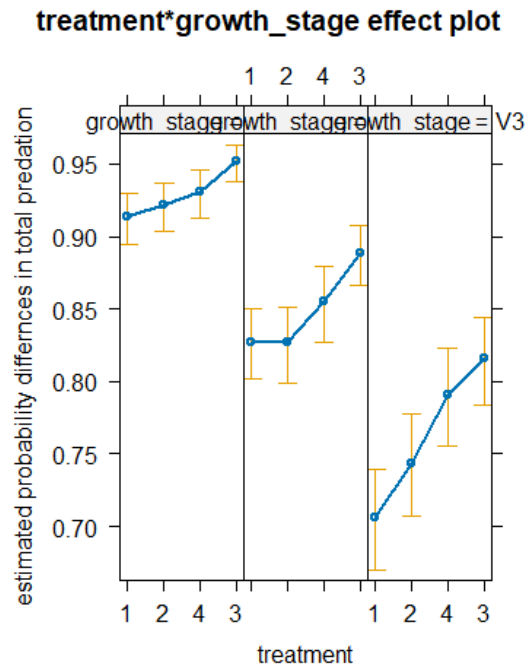
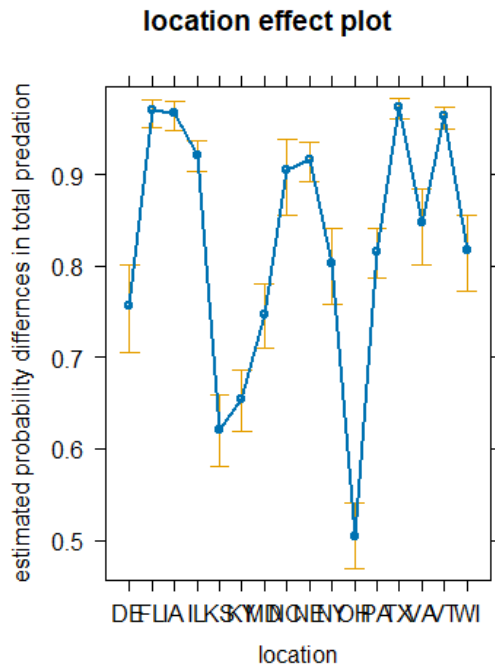
##	2.5 %	97.5 %
## (Intercept)	1.31014631	1.9764733
## treatment2	-0.20474489	0.4265235
## treatment4	-0.09190485	0.5865214
## treatment3	0.28509424	0.9858519
## growth_stageV5	-1.06824986	-0.5176834
## growth_stageV3	-1.76019155	-1.2074538
## locationFL	1.79804534	2.9686447
## locationIA	1.72951366	2.8716798
## locationIL	0.98862594	1.6819113
## locationKS	-0.95206550	-0.3347817
## locationKY	-0.80116140	-0.1994166
## locationMD	-0.37572258	0.2657333
## locationNC	0.60266086	1.6947494
## locationNE	0.87868947	1.6416022
## locationNY	-0.09940847	0.6423269
## locationOH	-1.41297909	-0.8182118
## locationPA	0.03731949	0.6679220
## locationTX	2.01669945	3.0240716

```
## locationVA          0.17516768  0.9962104
## locationVT          1.72113454  2.5936007
## locationWI         -0.01257680  0.7489879
## treatment2:growth_stageV5 -0.51520284  0.2824962
## treatment4:growth_stageV5 -0.46773420  0.3886933
## treatment3:growth_stageV5 -0.55865528  0.3148212
## treatment2:growth_stageV3 -0.32175280  0.4762538
## treatment4:growth_stageV3 -0.21790726  0.6313537
## treatment3:growth_stageV3 -0.45613053  0.4082442
```

```
plot(allEffects(nr_m1), type = 'link', ylab = 'estimated log differences in
total predation')
```



```
plot(allEffects(nr_m1), type = 'response', ylab = 'estimated probability
differences in total predation')
```



```
Anova(nr_m1)

## Analysis of Deviance Table (Type II tests)
##
## Response: to.predated
##
##          LR Chisq Df Pr(>Chisq)
## treatment      56.84  3  2.776e-12
## growth_stage   358.18  2  < 2.2e-16
## location     1392.68 15  < 2.2e-16
## treatment:growth_stage    2.73  6    0.8416

# Adding random effects
# This is now a random intercept, fixed slope model

m1 <- glmer(to.predated ~ treatment*growth_stage + location + (1|year),
family = binomial, data = sent)
summary(m1)

## Generalized linear mixed model fit by maximum likelihood (Laplace
## Approximation) [glmerMod]
## Family: binomial ( logit )
## Formula: to.predated ~ treatment * growth_stage + location + (1 | year)
## Data: sent
##
##      AIC      BIC   logLik deviance df.resid
##  7471.7  7671.4  -3707.9   7415.7     9199
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -10.9678    0.1288    0.2608    0.4923    1.9490
##
## Random effects:
##   Groups Name          Variance Std.Dev.
##   year   (Intercept) 0.04714  0.2171
## Number of obs: 9227, groups: year, 3
##
## Fixed effects:
##                                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)                      1.671105    0.213465   7.828 4.94e-15
## treatment2                       0.114824    0.161266   0.712 0.476454
## treatment4                       0.246685    0.173169   1.425 0.154291
## treatment3                       0.630504    0.178832   3.526 0.000422
## growth_stageV5                   -0.792380    0.140852  -5.626 1.85e-08
## growth_stageV3                   -1.469365    0.141386 -10.393 < 2e-16
## locationFL                       2.265534    0.298178   7.598 3.01e-14
## locationIA                       2.180264    0.291175   7.488 7.00e-14
## locationIL                       1.192947    0.181142   6.586 4.53e-11
## locationKS                       -0.775964    0.162522  -4.775 1.80e-06
## locationKY                       -0.630372    0.158533  -3.976 7.00e-05
## locationMD                       -0.237190    0.169864  -1.396 0.162607
## locationNC                       1.090066    0.279295   3.903 9.50e-05
## locationNE                       1.115260    0.198292   5.624 1.86e-08
## locationNY                       0.195255    0.191175   1.021 0.307094
## locationOH                       -1.241215    0.156815  -7.915 2.47e-15
## locationPA                       0.227405    0.165172   1.377 0.168581
## locationTX                       2.311086    0.260455   8.873 < 2e-16
## locationVA                       0.422000    0.213662   1.975 0.048259
## locationVT                       2.007101    0.225436   8.903 < 2e-16
## locationWI                       0.238376    0.199087   1.197 0.231173
## treatment2:growth_stageV5        -0.116894    0.204044  -0.573 0.566720
## treatment4:growth_stageV5        -0.028821    0.218914  -0.132 0.895257
## treatment3:growth_stageV5        -0.113658    0.223304  -0.509 0.610765
## treatment2:growth_stageV3         0.086373    0.204068   0.423 0.672107
## treatment4:growth_stageV3         0.211538    0.217053   0.975 0.329763
## treatment3:growth_stageV3        -0.009684    0.220906  -0.044 0.965033

coef(m1)$year

##      (Intercept) treatment2 treatment4 treatment3 growth_stageV5
## growth_stageV3
## 2021    1.787646    0.1148241    0.2466854    0.6305042    -0.7923797    -
## 1.469365
## 2022    1.848339    0.1148241    0.2466854    0.6305042    -0.7923797    -
## 1.469365
## 2023    1.375910    0.1148241    0.2466854    0.6305042    -0.7923797    -
## 1.469365
##      locationFL locationIA locationIL locationKS locationKY locationMD
## 2021    2.265534    2.180264    1.192947 -0.7759639 -0.6303725 -0.2371905
## 2022    2.265534    2.180264    1.192947 -0.7759639 -0.6303725 -0.2371905
```

```
## 2023 2.265534 2.180264 1.192947 -0.7759639 -0.6303725 -0.2371905
## locationNC locationNE locationNY locationOH locationPA locationTX
## 2021 1.090066 1.11526 0.1952546 -1.241215 0.2274051 2.311086
## 2022 1.090066 1.11526 0.1952546 -1.241215 0.2274051 2.311086
## 2023 1.090066 1.11526 0.1952546 -1.241215 0.2274051 2.311086
## locationVA locationVT locationWI treatment2:growth_stageV5
## 2021 0.4219997 2.007101 0.2383757 -0.1168941
## 2022 0.4219997 2.007101 0.2383757 -0.1168941
## 2023 0.4219997 2.007101 0.2383757 -0.1168941
## treatment4:growth_stageV5 treatment3:growth_stageV5
## 2021 -0.0288212 -0.1136579
## 2022 -0.0288212 -0.1136579
## 2023 -0.0288212 -0.1136579
## treatment2:growth_stageV3 treatment4:growth_stageV3
## 2021 0.08637314 0.211538
## 2022 0.08637314 0.211538
## 2023 0.08637314 0.211538
## treatment3:growth_stageV3
## 2021 -0.009684317
## 2022 -0.009684317
## 2023 -0.009684317
```

Anova(m1)

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
```

```
##
```

```
## Response: to.predated
```

```
## Chisq Df Pr(>Chisq)
## treatment 55.3990 3 5.644e-12
## growth_stage 316.0596 2 < 2.2e-16
## location 1012.8526 15 < 2.2e-16
## treatment:growth_stage 2.7172 6 0.8434
```

```
m2 <- glmer(to.predated ~ treatment*growth_stage + location + (1|year) +
(1|year:block) + (1|year:block:plot_id) , family = binomial, data = sent)
summary(m2)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
```

```
## Approximation) [glmerMod]
```

```
## Family: binomial ( logit )
```

```
## Formula: to.predated ~ treatment * growth_stage + location + (1 | year) +
```

```
## (1 | year:block) + (1 | year:block:plot_id)
```

```
## Data: sent
```

```
##
```

```
## AIC BIC logLik deviance df.resid
## 7448.4 7662.3 -3694.2 7388.4 9197
```

```
##
```

```
## Scaled residuals:
```

```
## Min 1Q Median 3Q Max
## -13.7104 0.1268 0.2626 0.4822 1.9925
```

```
##
```

```
## Random effects:
## Groups          Name          Variance Std.Dev.
## year:block:plot_id (Intercept) 0.04813  0.2194
## year:block       (Intercept) 0.04200  0.2049
## year             (Intercept) 0.05287  0.2299
## Number of obs: 9227, groups: year:block:plot_id, 149; year:block, 18;
year, 3
##
## Fixed effects:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)      1.72603    0.22709   7.601 2.95e-14
## treatment2        0.11579    0.16425   0.705 0.480833
## treatment4        0.23815    0.17563   1.356 0.175103
## treatment3        0.61525    0.18094   3.400 0.000673
## growth_stageV5    -0.82651    0.14198  -5.821 5.84e-09
## growth_stageV3    -1.50246    0.14278 -10.523 < 2e-16
## locationFL        2.26429    0.29801   7.598 3.01e-14
## locationIA        2.17017    0.29095   7.459 8.73e-14
## locationIL        1.18174    0.18324   6.449 1.13e-10
## locationKS       -0.68942    0.16578  -4.159 3.20e-05
## locationKY       -0.65385    0.15945  -4.101 4.12e-05
## locationMD       -0.27132    0.17216  -1.576 0.115034
## locationNC        1.10648    0.28029   3.948 7.89e-05
## locationNE        1.12135    0.20385   5.501 3.78e-08
## locationNY        0.25242    0.19773   1.277 0.201746
## locationOH       -1.27469    0.15791  -8.072 6.91e-16
## locationPA        0.21505    0.16604   1.295 0.195257
## locationTX        2.31122    0.26072   8.865 < 2e-16
## locationVA        0.40372    0.21493   1.878 0.060330
## locationVT        2.00152    0.22580   8.864 < 2e-16
## locationWI        0.20231    0.20125   1.005 0.314769
## treatment2:growth_stageV5 -0.12063    0.20540  -0.587 0.556994
## treatment4:growth_stageV5 -0.02170    0.22017  -0.099 0.921486
## treatment3:growth_stageV5 -0.11133    0.22429  -0.496 0.619625
## treatment2:growth_stageV3  0.10109    0.20581   0.491 0.623294
## treatment4:growth_stageV3  0.22909    0.21844   1.049 0.294287
## treatment3:growth_stageV3  0.01714    0.22211   0.077 0.938505
```

- 7) Start to work on a Table 1 that summarizes variables of interest, possibly by groups of interest. At a minimum, summarize the response variable, by a grouping variable if one exists.

```
# table as a proportion
all_locs <- sent %>%
  group_by(location, treatment, growth_stage) %>%
  dplyr::summarise(prop = mean(to.predated),
                    sd = sd(to.predated),
                    n = n(),
                    se = sd/sqrt(n))

by_trt <- sent %>%
```

```

group_by(treatment) %>%
  dplyr::summarise(prop = mean(to.predated),
                    sd = sd(to.predated),
                    n = n(),
                    se = sd/sqrt(n))

by_trt

## # A tibble: 4 × 5
##   treatment prop      sd      n      se
##   <fct>      <dbl> <dbl> <int>   <dbl>
## 1 1          0.764 0.425  2593 0.00835
## 2 2          0.791 0.407  2353 0.00839
## 3 4          0.803 0.398  1943 0.00902
## 4 3          0.849 0.359  2338 0.00741

by_growth <- sent %>%
  group_by(growth_stage) %>%
  dplyr::summarise(prop = mean(to.predated),
                    sd = sd(to.predated),
                    n = n(),
                    se = sd/sqrt(n))

by_growth

## # A tibble: 3 × 5
##   growth_stage prop      sd      n      se
##   <fct>      <dbl> <dbl> <int>   <dbl>
## 1 R3          0.894 0.308  3098 0.00554
## 2 V5          0.793 0.405  3248 0.00710
## 3 V3          0.708 0.455  2881 0.00847

```

7a) Comparing my old way of summary stats with yours. Which do I like more?

```

datasummary(treatment + growth_stage ~ to.predated, data = sent, output =
'markdown')

```

		to.predated
treatment	1	2593.00
	2	2353.00
	4	1943.00
	3	2338.00
growth_stage	R3	3098.00
	V5	3248.00
	V3	2881.00

I do not think this is great for binary data.

- 8) Provide the names of feedback group members and the date, time, and location of your feedback session interaction.

Graded for completion/not but there are points for participation in a feedback session.
Note that 412 students get full credit for this.