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.

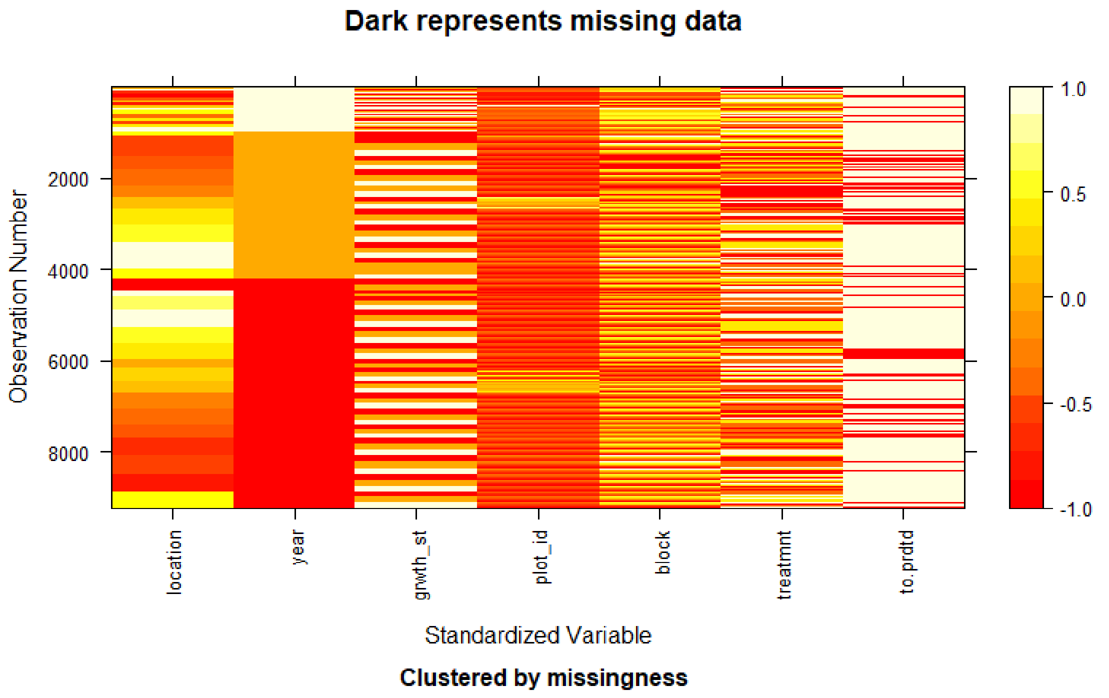
Study design: Treatments = 4; No cover crop, early-terminated cover crop, late-terminated cover crop, planting green **Plots** = 20; 5 blocks composed of 4 plots each = 20 plots / study site / year **Years** = 3 (2021,2022,2023) **Locations** = This project comprises 16 states. Not all states collected sentinel prey data every year. Each site year was in a different field. **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 a binomial of 6 traps/ counts per plot. Pseudoreplication is account for in the random term. **Explanatory** = Crop growth stage (timing, three levels) and treatment (four levels). I am not interested in the fixed effects of location. **Random effects** = Plot in block in location, in year. I want to account for pseduoreplication and all of the site/year combinations.

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)
```

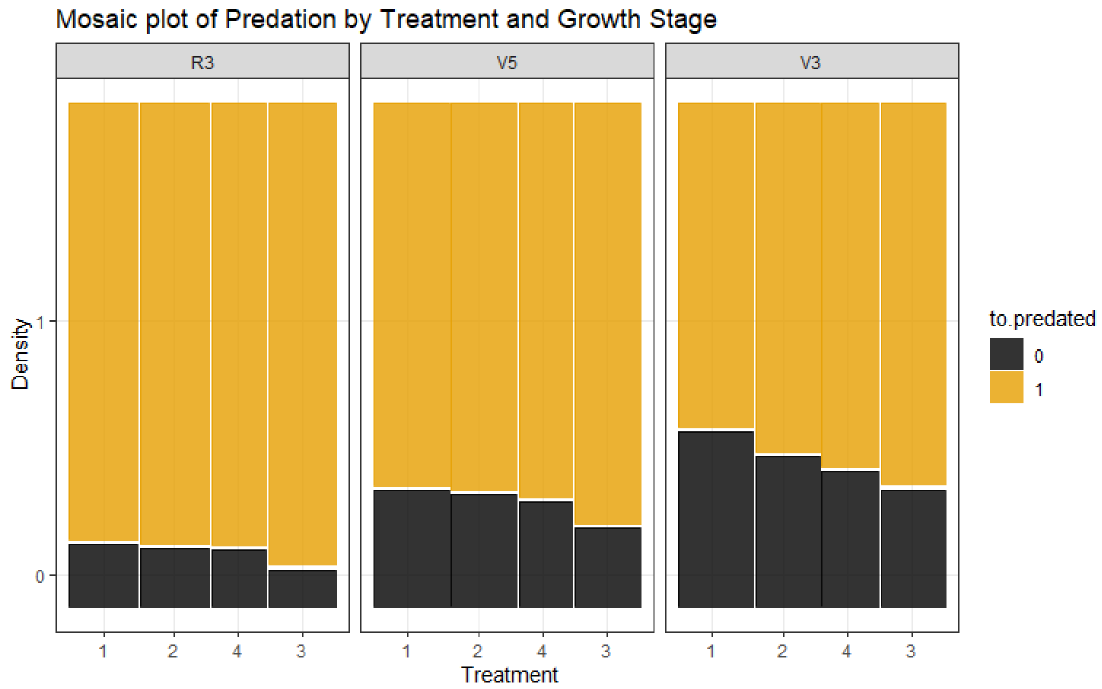


4) Discuss any other use in classes or theses for the data - either that you have used it for or are currently working on for future submissions.

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(treatment), fill = to.predated))+
  facet_wrap(~growth_stage)+
  scale_fill_colorblind()+
  labs(title = 'Mosaic plot of Predation by Treatment and Growth Stage',
       y = 'Density',
       x = ' Treatment')
```



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,family = binomial, data =
sent)
summary(nr_m1)$coefficients
```

```
##                              Estimate Std. Error     z value     Pr(>|z|)
## (Intercept)                1.95010303  0.1058235  18.4278755 7.850602e-76
## treatment2                 0.09001141  0.1523813   0.5906987 5.547223e-01
## treatment4                 0.11727980  0.1631881   0.7186786 4.723389e-01
## treatment3                 0.57828873  0.1708656   3.3844662 7.131680e-04
## growth_stageV5            -0.76347434  0.1307850  -5.8376279 5.294925e-09
## growth_stageV3            -1.33026838  0.1284608 -10.3554396 3.953661e-25
## treatment2:growth_stageV5 -0.03167141  0.1900681  -0.1666319 8.676597e-01
## treatment4:growth_stageV5  0.02841929  0.2041932   0.1391784 8.893092e-01
## treatment3:growth_stageV5 -0.07880787  0.2106201  -0.3741707 7.082773e-01
## treatment2:growth_stageV3  0.13678402  0.1876180   0.7290560 4.659674e-01
## treatment4:growth_stageV3  0.25462004  0.1998776   1.2738796 2.027061e-01
## treatment3:growth_stageV3 -0.01669189  0.2058425  -0.0810906 9.353699e-01
```

```
confint(nr_m1)
```

```
##                                2.5 %     97.5 %
## (Intercept)                1.7478677  2.1631305
## treatment2                -0.2085594  0.3895287
## treatment4                -0.2008588  0.4397899
## treatment3                 0.2467955  0.9177962
## growth_stageV5            -1.0228788 -0.5097501
## growth_stageV3            -1.5855908 -1.0816201
## treatment2:growth_stageV5 -0.4045729  0.3409235
## treatment4:growth_stageV5 -0.3730165  0.4279916
## treatment3:growth_stageV5 -0.4940689  0.3323383
## treatment2:growth_stageV3 -0.2313032  0.5045872
## treatment4:growth_stageV3 -0.1385465  0.6455444
## treatment3:growth_stageV3 -0.4229271  0.3847519
```

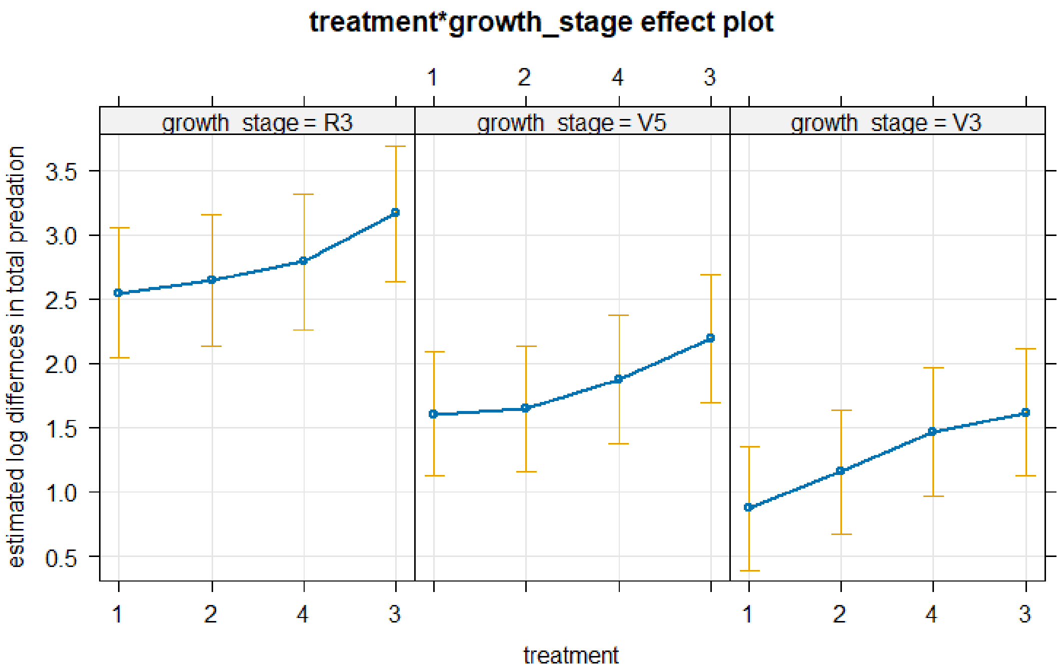
```
# Adding random effects
# This is now a random intercept, fixed slope model


m2 <- glmer(to.predated ~ treatment*growth_stage +
(1|year/location/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 + (1 | year/location/block/plot_id)
##    Data: sent
##
##      AIC      BIC   logLik deviance df.resid
##   7299.8   7413.8  -3633.9   7267.8     9211
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -10.6508   0.1008   0.2473   0.4551   2.5788
##
## Random effects:
##  Groups                            Name        Variance  Std.Dev.
##  plot_id:(block:(location:year)) (Intercept) 1.208e-01 0.3476046
##  block:(location:year)           (Intercept) 1.112e-01 0.3335262
##  location:year                   (Intercept) 1.928e+00 1.3886389
##  year                            (Intercept) 2.654e-08 0.0001629
## Number of obs: 9227, groups:
## plot_id:(block:(location:year)), 744; block:(location:year), 191;
location:year, 39; year, 3
##
## Fixed effects:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  2.54790    0.25831   9.864  < 2e-16
## treatment2                   0.09807    0.17162   0.571  0.56769
## treatment4                   0.24439    0.18491   1.322  0.18627
## treatment3                   0.61574    0.18791   3.277  0.00105
## growth_stageV5              -0.94486    0.14597  -6.473  9.6e-11
## growth_stageV3              -1.67496    0.14844 -11.283  < 2e-16
## treatment2:growth_stageV5 -0.05245    0.20982  -0.250  0.80260
## treatment4:growth_stageV5  0.02462    0.22579   0.109  0.91319
## treatment3:growth_stageV5 -0.02827    0.22772  -0.124  0.90121
## treatment2:growth_stageV3  0.18248    0.21183   0.861  0.38899
## treatment4:growth_stageV3  0.34828    0.22496   1.548  0.12158
## treatment3:growth_stageV3  0.12494    0.22719   0.550  0.58238
##
## Correlation of Fixed Effects:
##              (Intr) trtmn2 trtmn4 trtmn3 grw_V5 grw_V3 t2:_V5 t4:_V5 t3:_V5
## treatment2   -0.323
## treatment4   -0.296  0.450
## treatment3   -0.294  0.444  0.412
## grwth_stgV5 -0.366  0.532  0.491  0.486
## grwth_stgV3 -0.371  0.526  0.491  0.480  0.656
## trtmnt2:_V5  0.248 -0.763 -0.344 -0.338 -0.689 -0.450
## trtmnt4:_V5  0.230 -0.344 -0.756 -0.315 -0.640 -0.418  0.446
## trtmnt3:_V5  0.231 -0.341 -0.317 -0.776 -0.636 -0.417  0.442  0.411
## trtmnt2:_V3  0.248 -0.758 -0.341 -0.336 -0.454 -0.684  0.632  0.293  0.291
## trtmnt4:_V3  0.236 -0.346 -0.763 -0.316 -0.427 -0.650  0.297  0.633  0.274
## trtmnt3:_V3  0.235 -0.344 -0.317 -0.779 -0.425 -0.641  0.294  0.274  0.653
##             t2:_V3 t4:_V3
## treatment2
## treatment4
## treatment3
## grwth_stgV5
## grwth_stgV3
## trtmnt2:_V5
## trtmnt4:_V5
## trtmnt3:_V5
## trtmnt2:_V3
## trtmnt4:_V3  0.450
## trtmnt3:_V3  0.447  0.421
```

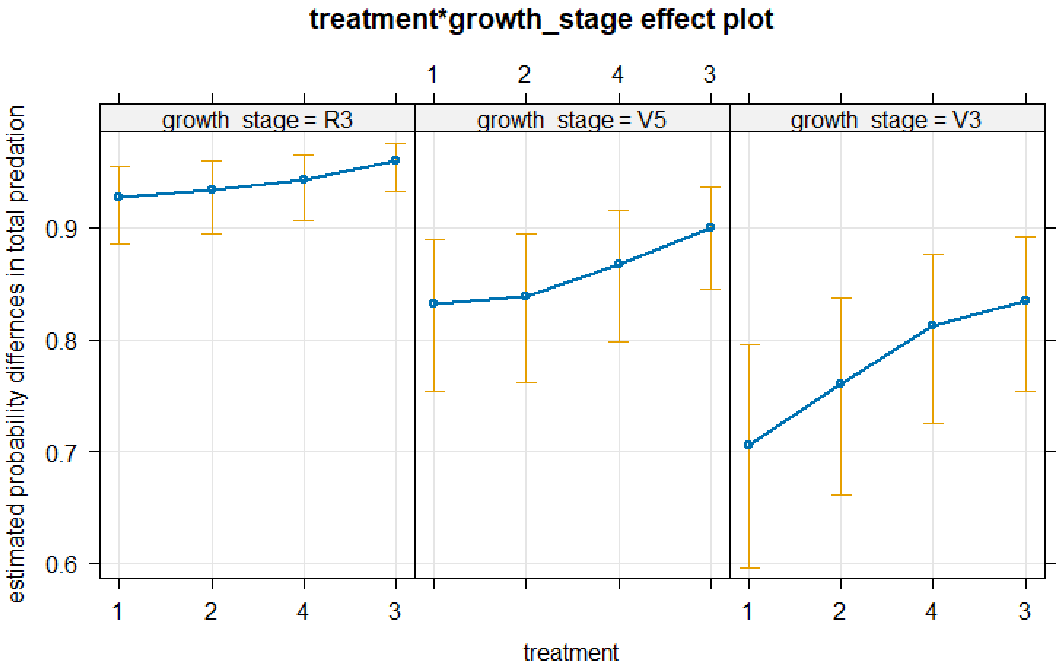
```
Anova(m2)
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: to.predated
##                            Chisq Df Pr(>Chisq)
## treatment               50.6763  3  5.734e-11
## growth_stage           342.8852  2  < 2.2e-16
## treatment:growth_stage   4.1298  6     0.6591
```

```
plot(allEffects(m2), type = 'link',ylab = 'estimated log differnces in total
predation', grid = T)
```



```
plot(allEffects(m2), type = 'response', ylab = 'estimated probability
differnces in total predation', grid = T)
```



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
sent %>%
  group_by(location, treatment, growth_stage) %>%
  summary()
```

```
##     location       year      growth_stage    plot_id       block     treatment
##  IL      : 946   2021:5004   R3:3098      103     : 450   1:2040   1:2593
##  OH      : 831   2022:3246   V5:3248      203     : 449   2:1845   2:2353
##  PA      : 825   2023: 977   V3:2881      303     : 447   3:1843   4:1943
##  KY      : 816                            401     : 443   4:1838   3:2338
##  VT      : 792                            101     : 434   5:1461
##  TX      : 720                            102     : 433   6: 200
##  (Other):4297                             (Other):6571
##   to.predated
##  Min.   :0.0000
##  1st Qu.:1.0000
##  Median :1.0000
##  Mean   :0.8005
##  3rd Qu.:1.0000
##  Max.   :1.0000
##
```

```
sent %>%
  group_by(growth_stage) %>%
  summary()
```

```
##     location       year      growth_stage    plot_id       block     treatment
##  IL      : 946   2021:5004   R3:3098      103     : 450   1:2040   1:2593
##  OH      : 831   2022:3246   V5:3248      203     : 449   2:1845   2:2353
##  PA      : 825   2023: 977   V3:2881      303     : 447   3:1843   4:1943
##  KY      : 816                            401     : 443   4:1838   3:2338
##  VT      : 792                            101     : 434   5:1461
##  TX      : 720                            102     : 433   6: 200
##  (Other):4297                             (Other):6571
##   to.predated
##  Min.   :0.0000
##  1st Qu.:1.0000
##  Median :1.0000
##  Mean   :0.8005
##  3rd Qu.:1.0000
##  Max.   :1.0000
##
```

```
tally(treatment ~ growth_stage, data = sent)
```

```
##          growth_stage
## treatment  R3  V5  V3
##         1 819 945 829
##         2 817 823 713
##         4 650 661 632
##         3 812 819 707
```

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.