EvansCode

Michelle Evans

Set-Up

NOTE: Working directory should be set to source file location.

```
knitr::opts_chunk$set(echo = T)
knitr::opts_chunk$set(cache=F)
knitr::opts_chunk$set(message=FALSE)
knitr::opts_chunk$set(warning=FALSE)
```

```
library(xtable)
library(tidyr)
library(ggplot2)
library(lme4) #mixed models
library(broom) #easy model comparison
library(MuMIn)
library(car)
library(MASS)
library(gridExtra) #for facet grids later
library(cowplot)
library(caret)
library(ggthemes)
library(survival)
library(DHARMa) #mixed model residuals
library(dplyr)
library(multcomp)

colR <- "dodgerblue"
colS <- "gray60"
colU <- "maroon"

axisColor <- errorColor <- "gray40"
```

Data Loading and Formatting

Load and format infection data:

```
formatData <- function(month){
  #' format infection data
  #' @params month (ie. "august")
  #' @returns dataframe of properly formatted data
  #adjust wingLength
  monthDf <- read.csv(paste0("../data/infections/raw/", month,"Dengue.csv"))
  #convert wingLength and drop extra columns
  monthDf$Wing <- monthDf$WingLength*monthDf$conversion..mm.bar.
  monthDf <- dplyr::select(monthDf, -WingLength, -conversion..mm.bar.)

  #dpi as factor
  monthDf$DPI <- as.factor(monthDf$DPI)

  #add in class and site
  monthDf$site <- as.factor(substr(as.character(monthDf$Individual), 1, 2))
  monthDf$class <- NULL
  for (i in 1:nrow(monthDf)){
    if (substr(monthDf$site[i], 1,1)=="R"){
    monthDf$class[i] <- "Rural"
    } else if (substr(monthDf$site[i], 1,1)=="S"){
    monthDf$class[i] <- "Suburban"
    } else if (substr(monthDf$site[i], 1,1)=="U"){
    monthDf$class[i] <- "Urban"
    }
  }
  monthDf$class <- as.factor(monthDf$class)

  #convert Y and N to 1 and 0 for statistics
  levels(monthDf$Body) <- c("NA", 0, 1)
  monthDf$Body <- as.numeric(as.character(monthDf$Body))
  levels(monthDf$Saliva) <- c("NA", 0, 1)
  monthDf$Saliva <- as.numeric(as.character(monthDf$Saliva))
  # august had no contaminated heads, so different corrections
  if (month=="august"){
    levels(monthDf$Head) <- c(0, 1)
  } else levels(monthDf$Head) <- c("NA",0, 1)
  monthDf$Head <- as.numeric(as.character(monthDf$Head))

  ##Fix false negatievs
  #adjust so that if saliva is positive, so is head
  #ddjust so that is head is positive, so is body
  monthDf$Head[monthDf$Saliva>0] <- 1
  monthDf$Body[monthDf$Head>0] <- 1

  return(monthDf)
}

august <- formatData("august")
oct <- formatData("october")
seasons <- rbind(august,oct)
seasons$block <- as.factor(c(rep("summer", nrow(august)), rep("fall", nrow(oct))))

seasonSumm <- seasons %>%
  filter(DPI==21) %>%
  #drop individual
  dplyr::select(-Individual, -site, -Wing, -DPI) %>%
  group_by(block, class) %>%
  summarise_all(funs(mean(.,na.rm=T),sd(.,na.rm=T),se=(sd(., na.rm=T)/sqrt(n())))) %>%
  ungroup()
```

Load and format microclimate data

```
climate <- read.csv('../data/microclimate/clean/2016TrialsAdult.csv')[,-1]
#toss out ridiculous levels
climate <- climate[climate$Temp<75,]
```

```
#format date
climate$Date <- strptime(climate$Date, format="%Y-%m-%d %H:%M:%S")
#draw out day
climate$Day <- as.Date(climate$Date)

# add tray id to climate data
trayID <- read.csv("../data/microclimate/trayLoggerID.csv") #read in IDs
climate <- merge(climate, trayID, by="Pot_ID")

#fix duplicates for R1T1
climate <- unique(climate)

#U2T2 and U1T2 are missing data
# range(climate[climate$Tray_ID=="U2T2", 'Date'])
# range(climate[climate$Tray_ID=="U1T2", 'Date'])

#drop U2T2 becuase it only has data until August 5th
inds <- which(climate$Tray_ID=="U2T2")
climate <- climate[-inds,]
rm(inds)

#U2T4 wasn't working right, reporting temps above 40C in October
inds <- which(climate$Tray_ID=="U2T4")
climate <- climate[-inds,]
rm(inds)
```

Load and format emergence data

Subset and standardize emergence data to be used with infections (i.e. weight by number of days mosquitoes were in larval environment)

Infection Dynamics

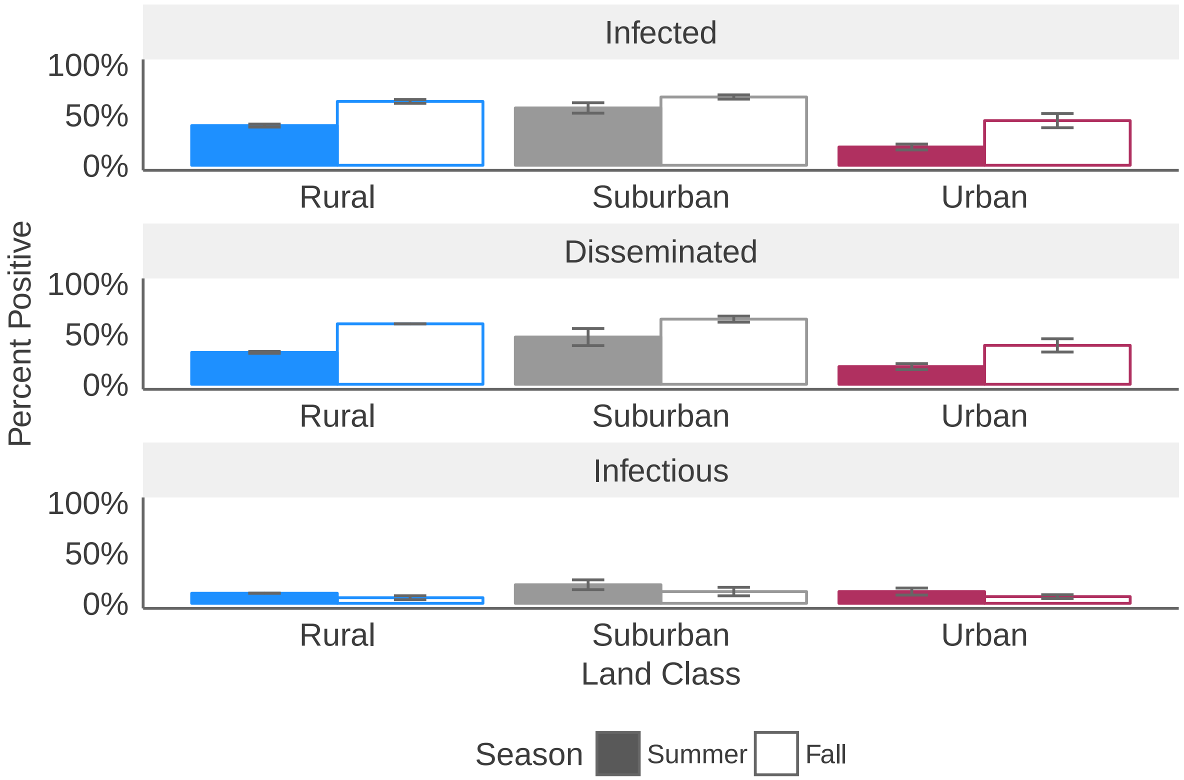
Infection by Class and Season

Make data long

```
infLong <- seasonInf %>%
  gather(type, infection, Body:Saliva) %>%
  dplyr::select(block, class, site, Individual, type, infection) %>%
  dplyr::group_by(block, class, type, site) %>%
  summarise(mean.inf=mean(infection, na.rm=T), sampleSize=sum(!is.na(infection)), positive=sum(infectio
  group_by(block, class, type) %>%
  summarise(se.inf=sd(mean.inf, na.rm=T)/n(), mean.inf=mean(mean.inf), samples=sum(sampleSize), positiv
  ungroup() %>%
  mutate(stripLabel=case_when(
    type=="Body" ~ "Infected",
    type=="Head" ~ "Disseminated",
    type=="Saliva" ~ "Infectious"
  ))
infLong$sampleLab <- paste0(infLong$positives, "(", infLong$samples, ")")
```

Plot (summer is alpha=1, fall is alpha=0.3)

```r
#pdf(file="figures/forMS/landclassXseasonInfection.pdf", width = 4, height=7,  family="sans")

ggplot(data=infLong[order(infLong$block, decreasing=F),], aes(x=class, group=block))+
  geom_bar(stat="identity", aes(y=mean.inf, alpha=block, fill=class, color=factor(class)), position=posi
  #geom_text(aes(label=sampleLab, y=mean.inf), vjust=-2, color="black", position=position_dodge(width=-
  facet_wrap(~factor(stripLabel, levels=c("Infected", "Disseminated", "Infectious")), nrow=3, dir="v", s
  scale_y_continuous(breaks=c(0,0.5,1), minor_breaks = c(0.25, 0.75), labels=c("0%", "50%", "100%"), lin
  #ylim(0,1)+
  scale_fill_manual(values=c(colR, colS, colU))+
  scale_color_manual(values=c(colR, colS, colU))+
  scale_alpha_discrete(range=c(0,1), name="Season", labels=c("Fall", "Summer"))+
  geom_errorbar(aes(ymin=mean.inf-se.inf, ymax=mean.inf+se.inf), width=0.2, color=errorColor, position =
  xlab("Land Class") +
  ylab("Percent Positive") +
  guides(fill=F, color= F,
         alpha=guide_legend(override.aes=list(color=axisColor), reverse=T)) +
  theme_fivethirtyeight() +
  theme(panel.background = element_rect(fill = "transparent", colour = NA),
        plot.background = element_rect(fill = "transparent", colour = NA),
        legend.key = element_blank(),
        panel.grid.major.x = element_blank(),
        legend.background = element_rect(fill = "transparent", colour = NA),
        axis.title = element_text(),
        axis.title.x = element_text(),
        axis.title.y=element_text(),
        #strip.background = element_blank(),
        strip.text.x = element_text(size=12),
        axis.line=element_line(color=axisColor, size=0.5),
        panel.grid = element_blank(),
        axis.text.y=element_text(size=12),
        axis.text.x=element_text(size=12))
```



```
#dev.off()
```

Statistics on infection by season and land class

Body model selection

```
#create all the models
m0 <- glmer(Body~1 + (1|site),
                          data=seasons[seasons$DPI=="21",],
                          family=binomial(link="logit"))
m1 <- glmer(Body~block + (1|site),
                          data=seasons[seasons$DPI=="21",],
                          family=binomial(link="logit"))
m2 <- glmer(Body~class +  (1|site),
                          data=seasons[seasons$DPI=="21",],
                          family=binomial(link="logit"))
m3 <- glmer(Body~class + block + (1|site),
                          data=seasons[seasons$DPI=="21",],
                          family=binomial(link="logit"))
m4 <- glmer(Body~class*block + (1|site),
                          data=seasons[seasons$DPI=="21",],
                          family=binomial(link="logit"))

modelSums <- do.call(rbind, lapply(list(m0,m1,m2,m3, m4), broom::glance)) #m3 is best

AICc(m0,m1,m2,m3,m4) #still 3
```

```
##    df     AICc
## m0  2 435.5883
## m1  3 426.0884
## m2  4 428.7311
## m3  5 417.7706
## m4  7 419.9622
```

```
Weights(AICc(m0,m1,m2,m3,m4))
```

```
## [1] 9.982942e-05 1.153806e-02 3.078070e-03 7.384377e-01 2.468464e-01
```

```
#create all the models
m0 <- glmer(Head~1 + (1|site),
                          data=seasons[seasons$DPI=="21",],
                          family=binomial(link="logit"))

m1 <- glmer(Head~block + (1|site),
                          data=seasons[seasons$DPI=="21",],
                          family=binomial(link="logit"))
m2 <- glmer(Head~class +  (1|site),
                          data=seasons[seasons$DPI=="21",],
                          family=binomial(link="logit"))
m3 <- glmer(Head~class + block + (1|site),
                          data=seasons[seasons$DPI=="21",],
                          family=binomial(link="logit"))
m4 <- glmer(Head~class*block + (1|site),
                          data=seasons[seasons$DPI=="21",],
                          family=binomial(link="logit"))

modelSums <- do.call(rbind, lapply(list(m0,m1,m2,m3, m4), broom::glance)) #model 3 is best

AICc(m0,m1,m2,m3,m4)
```

```
##    df     AICc
## m0  2 431.6814
## m1  3 419.6907
## m2  4 427.6540
## m3  5 414.1413
## m4  7 417.3718
```

```
tidy(m3)
```

```
##                   term      estimate std.error statistic      p.value group
## 1          (Intercept)  2.817563e-01 0.2342544  1.202779 0.2290617845 fixed
## 2        classSuburban  3.523897e-01 0.2867938  1.228721 0.2191763368 fixed
## 3           classUrban -7.675402e-01 0.2933131 -2.616795 0.0088759581 fixed
## 4          blocksummer -9.375460e-01 0.2416696 -3.879454 0.0001046913 fixed
## 5 sd_(Intercept).site  2.255814e-07        NA        NA           NA  site
```

```
#create all the models
m0 <- glmer(Saliva~1 + (1|site),
                          data=seasons[seasons$DPI=="21",],
                          family=binomial(link="logit"))
m1 <- glmer(Saliva~block + (1|site),
                          data=seasons[seasons$DPI=="21",],
                          family=binomial(link="logit"))
m2 <- glmer(Saliva~class +  (1|site),
                          data=seasons[seasons$DPI=="21",],
                          family=binomial(link="logit"))
```

```
m3 <- glmer(Saliva~class + block + (1|site),
                          data=seasons[seasons$DPI=="21",],
                          family=binomial(link="logit"))
m4 <- glmer(Saliva~class*block + (1|site),
                          data=seasons[seasons$DPI=="21",],
                          family=binomial(link="logit"))
m5 <- glmer(Saliva~ 1 + (1|site),
                          data=seasons[seasons$DPI=="21",],
                          family=binomial(link="logit")) #null model

modelSums <- do.call(rbind, lapply(list(m0, m1,m2,m3, m4, m5), broom::glance)) #model 1 is best (basica

AICc(m0, m1,m2,m3,m4, m5) #model 1
```

```
##    df     AICc
## m0  2 216.2245
## m1  3 214.1310
## m2  4 219.1221
## m3  5 217.1499
## m4  7 221.0126
## m5  2 216.2245
```

```
tidy(m1)
```

```
##                    term   estimate std.error statistic      p.value group
## 1           (Intercept) -2.6473436 0.3430712 -7.716600 1.194739e-14 fixed
## 2           blocksummer  0.7827283 0.3988372  1.962526 4.970127e-02 fixed
## 3 sd_(Intercept).site  0.1700891        NA        NA           NA  site
```

```
mixModelseasonsBody21 <- lme4::glmer(Body~class + block + (1|site),
                          data=seasons[seasons$DPI=="21",],
                          family=binomial(link="logit"))
summary(mixModelseasonsBody21)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Body ~ class + block + (1 | site)
##    Data: seasons[seasons$DPI == "21", ]
##
##      AIC      BIC   logLik deviance df.resid
##    417.6    436.4   -203.8    407.6      311
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.5779 -0.8367 -0.5458  0.9716  1.8323
##
## Random effects:
##  Groups Name        Variance  Std.Dev.
##  site   (Intercept) 6.524e-15 8.077e-08
## Number of obs: 316, groups:  site, 9
##
## Fixed effects:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    0.4917     0.2368   2.077 0.037846 *
## classSuburban   0.4205     0.2886   1.457 0.145130    
## classUrban     -0.8483     0.2892  -2.933 0.003360 ** 
## blocksummer    -0.8546     0.2407  -3.551 0.000384 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Correlation of Fixed Effects:
##             (Intr) clssSb clssUr
## classSubrbn -0.540              
## classUrban  -0.644  0.464       
## blocksummer -0.544 -0.069  0.123
```

```
tidy(mixModelseasonsBody21)
```

```
##                  term      estimate std.error statistic      p.value group
## 1         (Intercept)  4.917107e-01 0.2367960  2.076517 0.0378462050 fixed
## 2       classSuburban  4.205379e-01 0.2886423  1.456952 0.1451296729 fixed
## 3          classUrban -8.482715e-01 0.2892454 -2.932706 0.0033602215 fixed
## 4         blocksummer -8.546034e-01 0.2406755 -3.550854 0.0003839835 fixed
## 5 sd_(Intercept).site  8.076923e-08        NA        NA           NA  site
```

```
car::Anova(mixModelseasonsBody21) #Wald test
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
## 
## Response: Body
##        Chisq Df Pr(>Chisq)    
## class 18.733  2  8.553e-05 ***
## block 12.609  1   0.000384 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#confint(mixModelseasonsBody21) #profiled confidence interval
drop1(mixModelseasonsBody21, test="Chisq") #Likelihood ratio test
```

```
## Single term deletions
## 
## Model:
## Body ~ class + block + (1 | site)
##        Df    AIC    LRT   Pr(Chi)    
## <none>    417.58                     
## class   2 426.01 12.434 0.0019948 ** 
## block   1 428.60 13.025 0.0003073 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(multcomp::glht(mixModelseasonsBody21, linfct = multcomp::mcp(class = "Tukey"), test = multcomp:
```

```
## 
##   Simultaneous Tests for General Linear Hypotheses
## 
## Multiple Comparisons of Means: Tukey Contrasts
## 
## 
## Fit: lme4::glmer(formula = Body ~ class + block + (1 | site), data = seasons[seasons$DPI == 
##     "21", ], family = binomial(link = "logit"))
## 
```

```
## Linear Hypotheses:
##                          Estimate Std. Error z value Pr(>|z|)
## Suburban - Rural == 0      0.4205     0.2886   1.457  0.31179
## Urban - Rural == 0        -0.8483     0.2892  -2.933  0.00939 **
## Urban - Suburban == 0     -1.2688     0.2990  -4.243  < 0.001 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

```
summary(multcomp::glht(mixModelseasonsBody21, linfct = multcomp::mcp(block = "Tukey"), test = multcomp:
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: lme4::glmer(formula = Body ~ class + block + (1 | site), data = seasons[seasons$DPI ==
##     "21", ], family = binomial(link = "logit"))
##
## Linear Hypotheses:
##                       Estimate Std. Error z value Pr(>|z|)
## summer - fall == 0   -0.8546     0.2407  -3.551 0.000384 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

```
mixModelseasonsHead21 <- lme4::glmer(Head~class + block  + (1|site),
                          data=seasons[seasons$DPI=="21",],
                          family=binomial(link="logit"))
tidy(mixModelseasonsHead21)
```

```
##                  term      estimate std.error statistic      p.value group
## 1         (Intercept)  2.817563e-01 0.2342544  1.202779 0.2290617845 fixed
## 2       classSuburban  3.523897e-01 0.2867938  1.228721 0.2191763368 fixed
## 3          classUrban -7.675402e-01 0.2933131 -2.616795 0.0088759581 fixed
## 4         blocksummer -9.375460e-01 0.2416696 -3.879454 0.0001046913 fixed
## 5 sd_(Intercept).site  2.255814e-07        NA        NA           NA  site
```

```
car::Anova(mixModelseasonsHead21) #Wald test
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: Head
##        Chisq Df Pr(>Chisq)
## class 14.208  2  0.0008220 ***
## block 15.050  1  0.0001047 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#confint(mixModelseasonsHead21) #profiled confidence interval
drop1(mixModelseasonsHead21, test="Chisq")
```

```
## Single term deletions
##
## Model:
## Head ~ class + block + (1 | site)
##        Df    AIC     LRT  Pr(Chi)
## <none>    413.95
## class   2 419.61  9.6653 0.007966 **
## block   1 427.53 15.5772 7.92e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(multcomp::glht(mixModelseasonsHead21, linfct = multcomp::mcp(class = "Tukey"), test = multcomp:
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: lme4::glmer(formula = Head ~ class + block + (1 | site), data = seasons[seasons$DPI ==
##     "21", ], family = binomial(link = "logit"))
##
## Linear Hypotheses:
##                         Estimate Std. Error z value Pr(>|z|)
## Suburban - Rural == 0    0.3524     0.2868   1.229  0.43599
## Urban - Rural == 0      -0.7675     0.2933  -2.617  0.02397 *
## Urban - Suburban == 0  -1.1199     0.3031  -3.695  0.00065 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

```
summary(multcomp::glht(mixModelseasonsHead21, linfct = multcomp::mcp(block = "Tukey"), test = multcomp:
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: lme4::glmer(formula = Head ~ class + block + (1 | site), data = seasons[seasons$DPI ==
##     "21", ], family = binomial(link = "logit"))
##
## Linear Hypotheses:
##                      Estimate Std. Error z value Pr(>|z|)
## summer - fall == 0  -0.9375     0.2417  -3.879 0.000105 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

```
mixModelseasonsSaliva21 <- lme4::glmer(Saliva~block  + (1|site),
                          data=seasons[seasons$DPI=="21",],
                          family=binomial(link="logit"))
tidy(mixModelseasonsSaliva21) #z-test
```

```
##                  term   estimate std.error statistic      p.value group
## 1         (Intercept) -2.6473436 0.3430712 -7.716600 1.194739e-14 fixed
## 2         blocksummer  0.7827283 0.3988372  1.962526 4.970127e-02 fixed
## 3 sd_(Intercept).site  0.1700891        NA        NA           NA  site
```

```r
car::Anova((mixModelseasonsSaliva21)) #Wald test
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: Saliva
##        Chisq Df Pr(>Chisq)
## block 3.8515  1     0.0497 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
confint(mixModelseasonsSaliva21) #profiled confidence interval
```

```
##                   2.5 %     97.5 %
## .sig01       0.00000000  0.8214351
## (Intercept) -3.41631136 -2.0450739
## blocksummer  0.02729408  1.6074316
```

```r
drop1(mixModelseasonsSaliva21, test="Chisq")
```

```
## Single term deletions
##
## Model:
## Saliva ~ block + (1 | site)
##        Df    AIC    LRT Pr(Chi)
## <none>    214.06
## block   1 216.19 4.1318 0.04208 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
#pairwise comparison
summary(multcomp::glht(mixModelseasonsSaliva21, linfct = multcomp::mcp(block = "Tukey"), test = multcomp
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: lme4::glmer(formula = Saliva ~ block + (1 | site), data = seasons[seasons$DPI ==
##     "21", ], family = binomial(link = "logit"))
##
## Linear Hypotheses:
##                    Estimate Std. Error z value Pr(>|z|)
## summer - fall == 0   0.7827     0.3988   1.963   0.0497 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

Infection Efficiency

Format data for infection efficiency

```r
bodyEff <- seasons %>%
  filter(DPI==21) %>%
  #drop individual
  dplyr::select(-Individual, -site, -Wing, -DPI, -Head, -Saliva) %>%
  group_by(block, class) %>%
  summarise(bodyMean=mean(Body,na.rm=T),bodySE=(sd(Body, na.rm=T)/sqrt(n()))) %>%
  ungroup()

headEff <- seasons %>%
  filter(DPI==21) %>%
  filter(Body==1) %>%
  #drop individual
  dplyr::select(-Individual, -site, -Wing, -DPI, -Body, -Saliva) %>%
  group_by(block, class) %>%
  summarise(headMean=mean(Head,na.rm=T),headSE=(sd(Head, na.rm=T)/sqrt(n()))) %>%
  ungroup()

salEff <-  seasons %>%
  filter(DPI==21) %>%
  filter(Head==1) %>%
  #drop individual
  dplyr::select(-Individual, -site, -Wing, -DPI, -Head, -Body) %>%
  group_by(block, class) %>%
  summarise(salMean=mean(Saliva,na.rm=T),salSE=(sd(Saliva, na.rm=T)/sqrt(n()))) %>%
  ungroup()

#group together
allEff <- full_join(bodyEff, headEff, by=c("block", "class"))
allEff <- full_join(allEff, salEff, by=c("block", "class"))

library(tidyr)
meltMean <- allEff %>%
  select(block, class, contains("Mean")) %>%
  gather(key=variable, value=mean, -block, -class)

meltMean$type <- rep(c("Body", "Head", "Saliva"), each=6)

meltSE <- allEff %>%
  select(block, class, contains("SE")) %>%
  gather(variable, SE, -block, -class)

meltSE$type <- rep(c("Body", "Head", "Saliva"), each=6)

meltAll <- full_join(meltMean, meltSE, by=c("class", "block", "type")) %>%
  select(-variable.x, -variable.y)
```

Plot of infection efficiency

```
#supplemental plot
ggplot(data=meltAll[order(meltAll$block, decreasing=F),], aes(x=class, group=block))+
  geom_bar(stat="identity", aes(y=mean, alpha=factor(block), fill=factor(class)), color="gray20", posit
  facet_wrap(~type, nrow=3, dir="v") +
  scale_fill_manual(values=c(colR, colS, colU))+
  scale_alpha_discrete(range=c(1,0.3), name="Season", labels=c("Fall", "Summer"), guide=guide_legend(re
  geom_errorbar(aes(ymin=mean-SE, ymax=mean+SE), width=0.2, color="gray20", position =position_dodge(-0
  theme_base() +
  xlab("Land Class") +
  ylab("Infection Efficiency") +
  guides(fill=F)
  #theme(legend.title=element_text("Season"))
```

Model Selection (Body is same as initial infection)

```
#create all the models
m1 <- glmer(Head~block + (1|site),
                          data=seasons[seasons$DPI=="21" & seasons$Body==1,],
                          family=binomial(link="logit"))
m2 <- glmer(Head~class +  (1|site),
                          data=seasons[seasons$DPI=="21" & seasons$Body==1,],
                          family=binomial(link="logit"))
m3 <- glmer(Head~class + block + (1|site),
                          data=seasons[seasons$DPI=="21" & seasons$Body==1,],
                          family=binomial(link="logit"))
m4 <- glmer(Head~class*block + (1|site),
                          data=seasons[seasons$DPI=="21" & seasons$Body==1,],
                          family=binomial(link="logit"))
m5 <- glmer(Head~class*(1|site),
                          data=seasons[seasons$DPI=="21" & seasons$Body==1,],
                          family=binomial(link="logit"))

modelSums <- do.call(rbind, lapply(list(m1,m2,m3, m4, m5), broom::glance)) #model 1 is best

AICc(m1,m2,m3,m4, m5)
```

```
##    df     AICc
## m1  3 106.3355
## m2  4 109.4356
## m3  5 110.3993
## m4  7 108.9752
## m5  4 109.4356
```

```
tidy(m1)
```

```
##                   term   estimate std.error statistic      p.value group
## 1          (Intercept)  2.3978953 0.3947713  6.074137 1.246562e-09 fixed
## 2          blocksummer -0.5877867 0.5339828 -1.100760 2.710013e-01 fixed
## 3 sd_(Intercept).site  0.0000000        NA        NA           NA  site
```

```
#create all the models
m1 <- glmer(Saliva~block + (1|site),
                          data=seasons[seasons$DPI=="21" & seasons$Body==1,],
                          family=binomial(link="logit"))
m2 <- glmer(Saliva~class +  (1|site),
                          data=seasons[seasons$DPI=="21" & seasons$Body==1,],
                          family=binomial(link="logit"))
m3 <- glmer(Saliva~class + block + (1|site),
                          data=seasons[seasons$DPI=="21" & seasons$Body==1,],
                          family=binomial(link="logit"))
m4 <- glmer(Saliva~class*block + (1|site),
                          data=seasons[seasons$DPI=="21" & seasons$Body==1,],
                          family=binomial(link="logit"))
m5 <- glmer(Saliva~class*(1|site),
                          data=seasons[seasons$DPI=="21" & seasons$Body==1,],
                          family=binomial(link="logit"))
```

```r
modelSums <- do.call(rbind, lapply(list(m1,m2,m3, m4, m5), broom::glance)) #between 1 and 3

AICc(m1,m2,m3,m4, m5)
```

```
##    df     AICc
## m1  3 150.9633
## m2  4 162.7401
## m3  5 150.3630
## m4  7 153.2814
## m5  4 162.7401
```

```r
tidy(m1)
```

```
##                  term   estimate std.error statistic      p.value group
## 1         (Intercept) -2.0361094 0.3666955 -5.552589 2.814701e-08 fixed
## 2         blocksummer  1.4633691 0.4480834  3.265841 1.091397e-03 fixed
## 3 sd_(Intercept).site  0.2624814        NA        NA           NA  site
```

```r
tidy(m3)
```

```
##                  term   estimate std.error  statistic      p.value group
## 1         (Intercept) -2.5015553 0.4924147 -5.0801796 3.770782e-07 fixed
## 2       classSuburban  0.1556519 0.5058740  0.3076891 7.583189e-01 fixed
## 3          classUrban  1.1685198 0.5635947  2.0733338 3.814123e-02 fixed
## 4         blocksummer  1.6490627 0.4616055  3.5724499 3.536572e-04 fixed
## 5 sd_(Intercept).site  0.0000000        NA         NA           NA  site
```

```r
#model
mixModelseasonsBody21 <- lme4::glmer(Body~class + block + (1|site),
                          data=seasons[seasons$DPI=="21",],
                          family=binomial(link="logit"))
tidy(mixModelseasonsBody21)
```

```
##                  term      estimate std.error statistic      p.value group
## 1         (Intercept)  4.917107e-01 0.2367960  2.076517 0.0378462050 fixed
## 2       classSuburban  4.205379e-01 0.2886423  1.456952 0.1451296729 fixed
## 3          classUrban -8.482715e-01 0.2892454 -2.932706 0.0033602215 fixed
## 4         blocksummer -8.546034e-01 0.2406755 -3.550854 0.0003839835 fixed
## 5 sd_(Intercept).site  8.076923e-08        NA        NA           NA  site
```

```r
car::Anova(mixModelseasonsBody21)
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: Body
##        Chisq Df Pr(>Chisq)
## class 18.733  2  8.553e-05 ***
## block 12.609  1   0.000384 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
summary(multcomp::glht(mixModelseasonsBody21, linfct = multcomp::mcp(class = "Tukey"), test = multcomp:
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Tukey Contrasts
```

```
##
##
## Fit: lme4::glmer(formula = Body ~ class + block + (1 | site), data = seasons[seasons$DPI ==
##     "21", ], family = binomial(link = "logit"))
##
## Linear Hypotheses:
##                         Estimate Std. Error z value Pr(>|z|)
## Suburban - Rural == 0     0.4205     0.2886   1.457  0.31182
## Urban - Rural == 0       -0.8483     0.2892  -2.933  0.00932 **
## Urban - Suburban == 0    -1.2688     0.2990  -4.243  < 0.001 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

```
summary(multcomp::glht(mixModelseasonsBody21, linfct = multcomp::mcp(block = "Tukey"), test = multcomp:
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: lme4::glmer(formula = Body ~ class + block + (1 | site), data = seasons[seasons$DPI ==
##     "21", ], family = binomial(link = "logit"))
##
## Linear Hypotheses:
##                      Estimate Std. Error z value Pr(>|z|)
## summer - fall == 0   -0.8546     0.2407  -3.551 0.000384 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

```
mixModelseasonsHead21 <- lme4::glmer(Head~block  + (1|site),
                          data=seasons[seasons$DPI=="21" & seasons$Body==1,],
                          family=binomial(link="logit"))
tidy(mixModelseasonsHead21)
```

```
##                   term   estimate std.error statistic      p.value group
## 1          (Intercept)  2.3978953 0.3947713  6.074137 1.246562e-09 fixed
## 2          blocksummer -0.5877867 0.5339828 -1.100760 2.710013e-01 fixed
## 3 sd_(Intercept).site  0.0000000        NA        NA           NA  site
```

```
car::Anova(mixModelseasonsHead21) #no effect
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: Head
##        Chisq Df Pr(>Chisq)
## block 1.2117  1      0.271
```

```
drop1(mixModelseasonsHead21, test="Chisq")
```

```
## Single term deletions
##
## Model:
## Head ~ block + (1 | site)
##        Df    AIC    LRT Pr(Chi)
## <none>    106.17
## block   1 105.39 1.2235  0.2687
```

```
summary(multcomp::glht(mixModelseasonsHead21, linfct = multcomp::mcp(block = "Tukey"), test = multcomp:
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: lme4::glmer(formula = Head ~ block + (1 | site), data = seasons[seasons$DPI ==
##     "21" & seasons$Body == 1, ], family = binomial(link = "logit"))
##
## Linear Hypotheses:
##                      Estimate Std. Error z value Pr(>|z|)
## summer - fall == 0  -0.5878     0.5340  -1.101    0.271
## (Adjusted p values reported -- single-step method)
```

```
mixModelseasonsSaliva21 <- lme4::glmer(Saliva~class + block  + (1|site),
                          data=seasons[seasons$DPI=="21" & seasons$Head==1,],
                          family=binomial(link="logit"))
tidy(mixModelseasonsSaliva21)
```

```
##                   term      estimate std.error  statistic      p.value
## 1          (Intercept) -2.378555e+00 0.4916104 -4.8382922 1.309595e-06
## 2        classSuburban  2.621010e-01 0.5198468  0.5041889 6.141286e-01
## 3           classUrban  1.122170e+00 0.5732326  1.9576171 5.027495e-02
## 4          blocksummer  1.714163e+00 0.4638138  3.6957993 2.191961e-04
## 5 sd_(Intercept).site  2.523364e-08        NA         NA           NA
##   group
## 1 fixed
## 2 fixed
## 3 fixed
## 4 fixed
## 5  site
```

```
car::Anova((mixModelseasonsSaliva21)) #only block is significant
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: Saliva
##         Chisq Df Pr(>Chisq)
## class  4.1137  2  0.1278553
## block 13.6589  1  0.0002192 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
drop1(mixModelseasonsSaliva21, test="Chisq")
```

```
## Single term deletions
##
## Model:
## Saliva ~ class + block + (1 | site)
##        Df    AIC    LRT   Pr(Chi)
## <none>    139.79
## class   2 139.95  4.1567    0.1251
## block   1 153.23 15.4317 8.554e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(multcomp::glht(mixModelseasonsSaliva21, linfct = multcomp::mcp(class = "Tukey"), test = multcomp
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: lme4::glmer(formula = Saliva ~ class + block + (1 | site), data = seasons[seasons$DPI ==
##     "21" & seasons$Head == 1, ], family = binomial(link = "logit"))
##
## Linear Hypotheses:
##                         Estimate Std. Error z value Pr(>|z|)
## Suburban - Rural == 0     0.2621     0.5198   0.504    0.869
## Urban - Rural == 0        1.1222     0.5732   1.958    0.123
## Urban - Suburban == 0     0.8601     0.5521   1.558    0.264
## (Adjusted p values reported -- single-step method)
```

```
summary(multcomp::glht(mixModelseasonsSaliva21, linfct = multcomp::mcp(block = "Tukey"), test = multcomp
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: lme4::glmer(formula = Saliva ~ class + block + (1 | site), data = seasons[seasons$DPI ==
##     "21" & seasons$Head == 1, ], family = binomial(link = "logit"))
##
## Linear Hypotheses:
##                      Estimate Std. Error z value Pr(>|z|)
## summer - fall == 0     1.7142     0.4638   3.696 0.000219 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

Infection and Microclimate

```
seasonSite <- seasons %>%
  filter(DPI=="21") %>%
  dplyr::select(-Individual, -DPI, - Wing) %>%
  dplyr::group_by(block, class, site) %>%
  summarise_all(funs(mean(.,na.rm=T), se=(sd(., na.rm=T)/sqrt(n()))))

#group with temperature data
seasonInfSite <- merge(seasonSite, rbind(augEnvVar,octEnvVar), by.x=c("block", "site"), by.y=c("block",

# inspect variable correlations
# cor(seasonInfSite[,10:18])
```

```
respV <- "Body_mean"
predVs <- c("Temp_mean_mean","RH_mean_mean", "DTR_mean", "Temp_min_mean", "RH_min_mean", "Temp_max_mean

myCols <- c(respV, predVs, "block","class", "site")
modDF <- seasonInfSite %>%
  dplyr::select(one_of(myCols))
modDF <- na.omit(modDF)

#model selection for initial variable (bc of high correlation)
m1 <- glm(Body_mean ~ Temp_mean_mean,
          data=modDF,
          family=gaussian(link="log"))
m2 <- glm(Body_mean ~ RH_mean_mean,
          data=modDF,
          family=gaussian(link="log"))
m3 <- glm(Body_mean ~ DTR_mean,
          data=modDF,
          family=gaussian(link="log"))
m4 <- glm(Body_mean ~ Temp_min_mean,
          data=modDF,
          family=gaussian(link="log"))
m5 <- glm(Body_mean ~ Temp_max_mean,
          data=modDF,
          family=gaussian(link="log"))
m6 <- glm(Body_mean ~ RH_min_mean,
          data=modDF,
          family=gaussian(link="log"))
m7 <- glm(Body_mean ~ RH_max_mean,
          data=modDF,
          family=gaussian(link="log"))

modelSums <- do.call(rbind, lapply(list(m1,m2,m3, m4, m5, m6, m7), broom::glance)) #1 seems best

AICc(m1,m2,m3, m4, m5, m6, m7)
```

```
##    df       AICc
## m1  3 -9.0569871
## m2  3 -0.5439088
## m3  3 -6.7374663
## m4  3 -8.0859090
## m5  3 -6.4929200
## m6  3 -2.3719484
## m7  3 -1.1379604
```

```
tidy(m1)
```

```
##             term    estimate  std.error statistic     p.value
## 1    (Intercept)  0.95866798 0.53632500  1.787476 0.092815805
## 2 Temp_mean_mean -0.07494337 0.02488645 -3.011412 0.008279968
```

```
#find covariates that aren't correlated
covars <- data.frame(cor(seasonInfSite[,10:18]))
rownames(covars[abs(covars$Temp_mean_mean)<0.8,]) #only RH_mean and RH_max
```

```
## [1] "RH_mean_mean" "RH_max_mean"  "Temp_mean_se" "RH_mean_se"
```

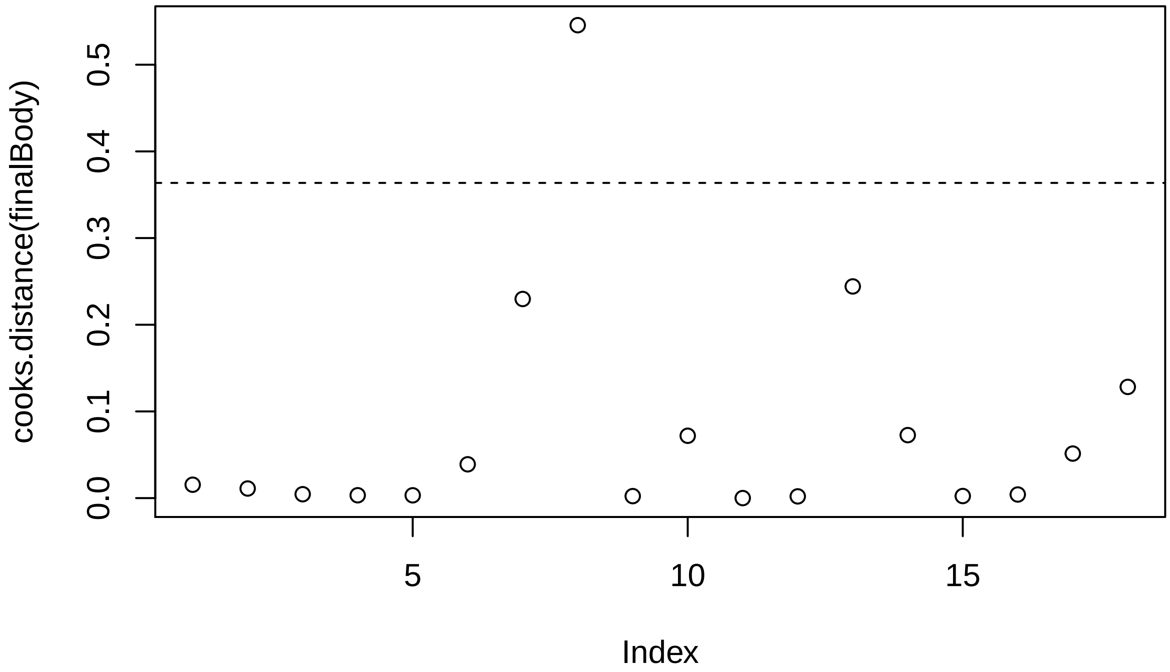
```
#model selection
m1 <- glm(Body_mean ~ Temp_mean_mean,
          data=modDF,
          family=gaussian(link="log"))
m2 <- glm(Body_mean ~ RH_mean_mean,
          data=modDF,
          family=gaussian(link="log"))

m3 <- glm(Body_mean ~ Temp_mean_mean+RH_mean_mean,
          data=modDF,
          family=gaussian(link="log"))
m4 <- glm(Body_mean ~ Temp_mean_mean*RH_mean_mean,
          data=modDF,
          family=gaussian(link="log"))




modelSums <- do.call(rbind, lapply(list(m1,m2, m3, m4), broom::glance)) #3 seems best

AICc(m1,m2, m3, m4)
```

```
##    df        AICc
## m1  3  -9.0569871
## m2  3  -0.5439088
## m3  4 -12.8120279
## m4  5  -9.0366850
```

```
tidy(m3)
```

```
##             term    estimate  std.error statistic      p.value
## 1    (Intercept) -1.02698904 0.88089922 -1.165842 0.2618821166
## 2 Temp_mean_mean -0.12883362 0.03098415 -4.158050 0.0008413049
## 3   RH_mean_mean  0.03760852 0.01518724  2.476323 0.0256769139
```

```
finalBody <- glm(Body_mean ~ Temp_mean_mean+RH_mean_mean,
          data=modDF,
          family=gaussian(link="log"))
plot(cooks.distance(finalBody))
abline(h=4/length(modDF), lty=2)
```



```
car::Anova(finalBody)
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: Body_mean
##                LR Chisq Df Pr(>Chisq)
## Temp_mean_mean  20.7454  1  5.246e-06 ***
## RH_mean_mean     7.2752  1   0.006991 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#confint(finalBody)
tidy(finalBody)
```

```
##              term    estimate  std.error statistic      p.value
## 1    (Intercept) -1.02698904 0.88089922 -1.165842 0.2618821166
## 2 Temp_mean_mean -0.12883362 0.03098415 -4.158050 0.0008413049
## 3   RH_mean_mean  0.03760852 0.01518724  2.476323 0.0256769139
```

```
drop1(finalBody, test="F")
```

```
## Single term deletions
##
## Model:
## Body_mean ~ Temp_mean_mean + RH_mean_mean
##                Df Deviance      AIC F value    Pr(>F)
## <none>              0.27952 -15.8890
## Temp_mean_mean  1  0.66611  -2.2582 20.7455 0.0003796 ***
## RH_mean_mean    1  0.41510 -10.7713  7.2753 0.0165491 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
outlierTest(finalBody)
```

```
##
```

```
## No Studentized residuals with Bonferonni p < 0.05
## Largest |rstudent|:
##   rstudent unadjusted p-value Bonferonni p
## 8 2.598889          0.0093526      0.16835
```

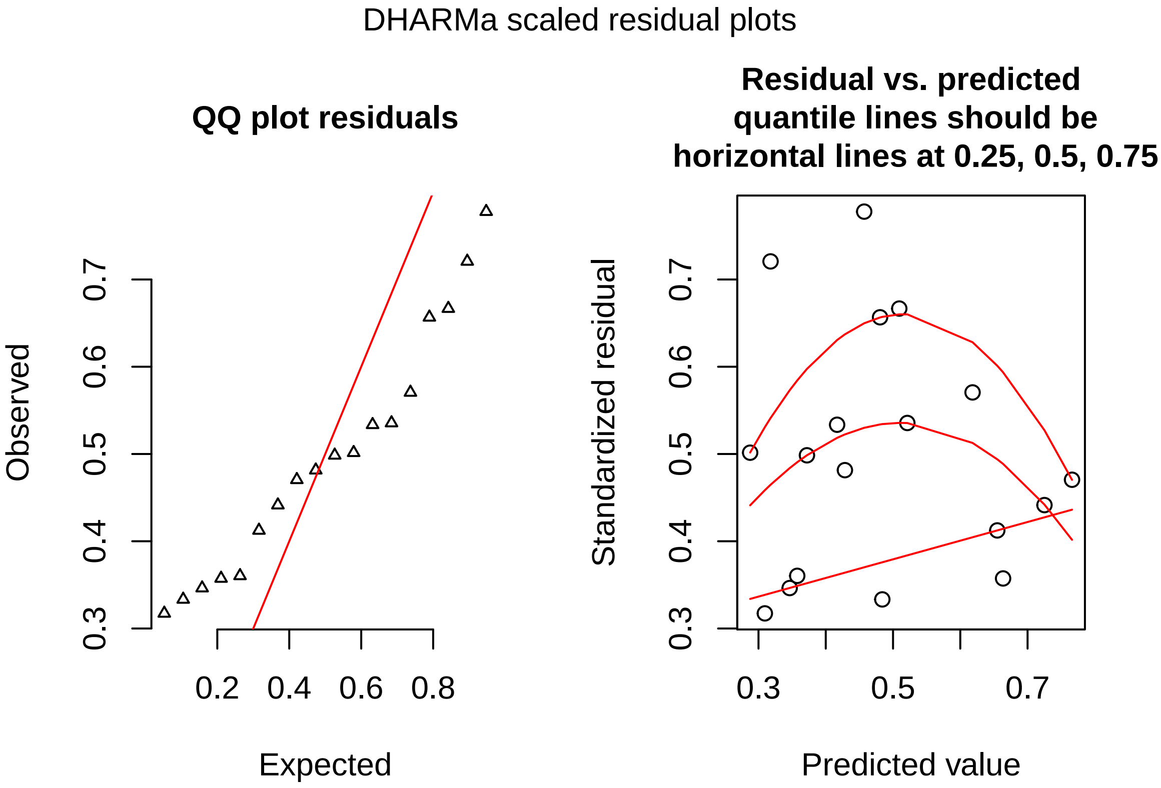
```
vif(finalBody)
```

```
## Temp_mean_mean   RH_mean_mean
##       2.500624       2.500624
```

```
sqrt(vif(finalBody))>2
```

```
## Temp_mean_mean   RH_mean_mean
##          FALSE          FALSE
```

```
#glm residuals
simulationOutput <- simulateResiduals(fittedModel = finalBody, n=999)
plotSimulatedResiduals(simulationOutput = simulationOutput)
```



```
respV <- "Head_mean"
predVs <- c("Temp_mean_mean","RH_mean_mean", "DTR_mean", "Temp_min_mean", "RH_min_mean", "Temp_max_mean

myCols <- c(respV, predVs, "block","class", "site")
modDF <- seasonInfSite %>%
  dplyr::select(one_of(myCols))
modDF <- na.omit(modDF)

#model selection for initial variable (bc of high correlation)
m1 <- glm(Head_mean ~ Temp_mean_mean,
          data=modDF,
          family=gaussian(link="log"))
```

```r
m2 <- glm(Head_mean ~ RH_mean_mean,
          data=modDF,
          family=gaussian(link="log"))
m3 <- glm(Head_mean ~ DTR_mean,
          data=modDF,
          family=gaussian(link="log"))
m4 <- glm(Head_mean ~ Temp_min_mean,
          data=modDF,
          family=gaussian(link="log"))
m5 <- glm(Head_mean ~ Temp_max_mean,
          data=modDF,
          family=gaussian(link="log"))
m6 <- glm(Head_mean ~ RH_min_mean,
          data=modDF,
          family=gaussian(link="log"))
m7 <- glm(Head_mean ~ RH_max_mean,
          data=modDF,
          family=gaussian(link="log"))

modelSums <- do.call(rbind, lapply(list(m1,m2,m3, m4, m5, m6, m7), broom::glance)) #1 seems best
tidy(m1)
```

```
##              term    estimate  std.error statistic     p.value
## 1     (Intercept)  1.24680019 0.59855991  2.083000 0.053652279
## 2 Temp_mean_mean -0.09311821 0.02822372 -3.299289 0.004526073
```

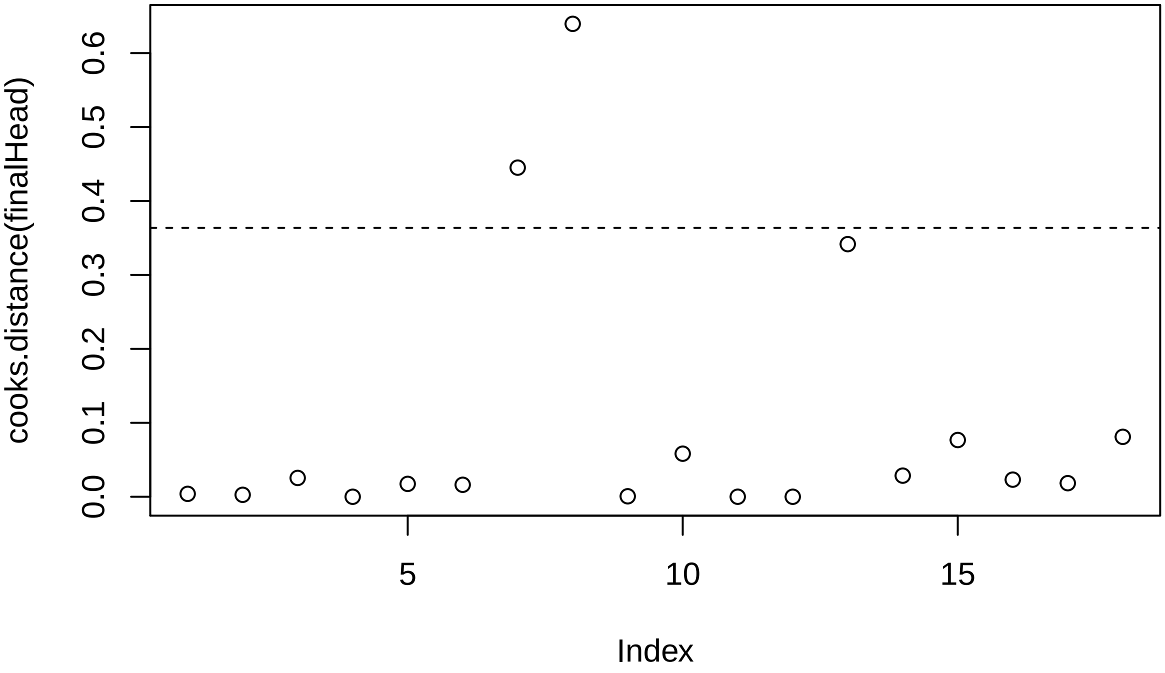
```r
#model selection
m1 <- glm(Head_mean ~ Temp_mean_mean,
          data=modDF,
          family=gaussian(link="log"))
m2 <- glm(Head_mean ~ RH_mean_mean,
          data=modDF,
          family=gaussian(link="log"))

m3 <- glm(Head_mean ~ Temp_mean_mean+RH_mean_mean,
          data=modDF,
          family=gaussian(link="log"))
m4 <- glm(Head_mean ~ Temp_mean_mean*RH_mean_mean,
          data=modDF,
          family=gaussian(link="log"))

modelSums <- do.call(rbind, lapply(list(m1,m2, m3, m4), broom::glance)) #3 seems best

AICc(m1,m2, m3, m4)
```

```
##    df         AICc
## m1  3  -9.81626751
## m2  3   0.06694249
## m3  4 -12.42640583
## m4  5  -8.51958039
```

```r
tidy(m3)
```

```
##              term    estimate  std.error  statistic      p.value
## 1     (Intercept) -0.76290213 1.00469599 -0.7593363 0.4594190579
## 2 Temp_mean_mean -0.14570591 0.03490278 -4.1746212 0.0008135761
## 3   RH_mean_mean  0.03758149 0.01708945  2.1991047 0.0439706330
```

```
finalHead <- glm(Head_mean ~ Temp_mean_mean+RH_mean_mean,
          data=modDF,
          family=gaussian)

plot(cooks.distance(finalHead))
abline(h=4/length(modDF), lty=2)
```



```
car::Anova(finalHead)
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: Head_mean
##                LR Chisq Df Pr(>Chisq)
## Temp_mean_mean  23.2199  1  1.445e-06 ***
## RH_mean_mean     7.3387  1   0.006749 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#confint(finalHead)
drop1(finalHead, test="F")
```

```
## Single term deletions
##
## Model:
## Head_mean ~ Temp_mean_mean + RH_mean_mean
##                Df Deviance      AIC F value    Pr(>F)
## <none>              0.26932 -16.5584
## Temp_mean_mean  1   0.68622  -1.7229 23.2199 0.0002255 ***
## RH_mean_mean    1   0.40108 -11.3895  7.3387 0.0161642 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

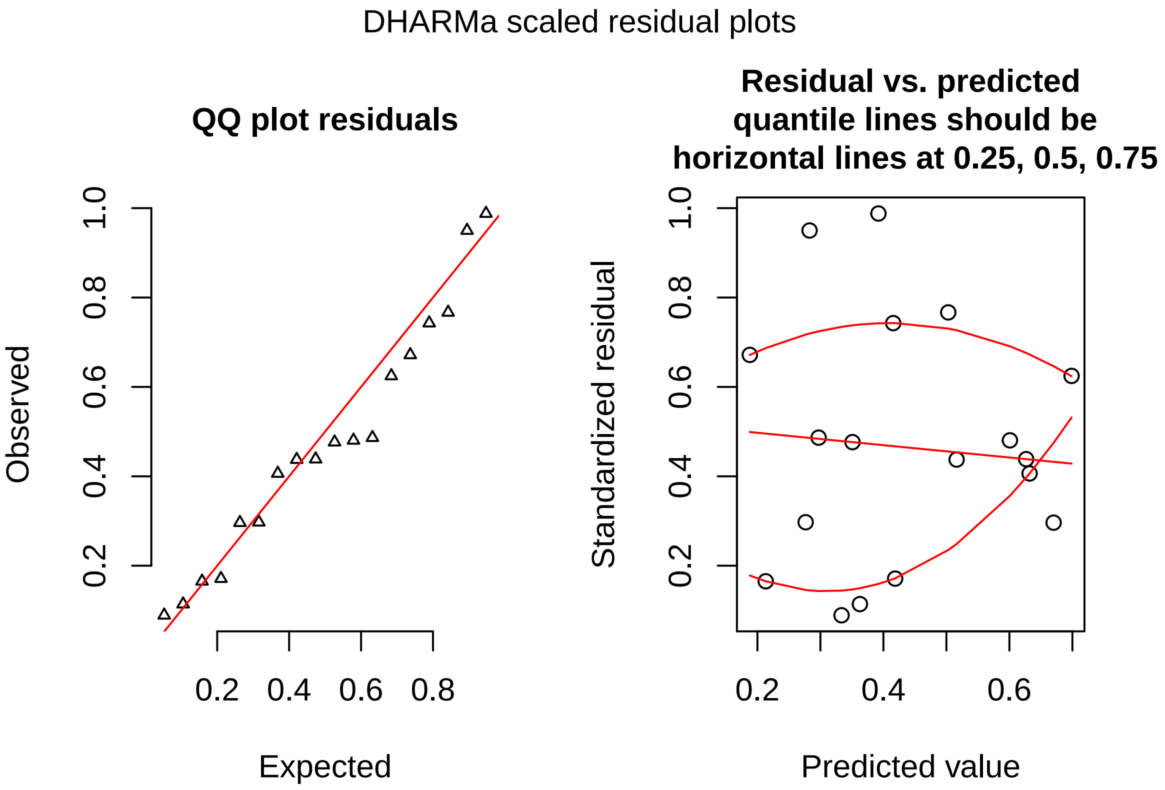
```
tidy(finalHead)
```

```
##              term    estimate  std.error statistic      p.value
## 1     (Intercept)  0.59188906 0.35139747  1.684386 0.1127946327
## 2 Temp_mean_mean -0.06672422 0.01384692 -4.818705 0.0002254676
## 3    RH_mean_mean  0.01603968 0.00592088  2.709003 0.0161642326
```

```
outlierTest(finalHead)
```

```
##    rstudent unadjusted p-value Bonferonni p
## 13 3.213467          0.0013114     0.023606
```

```
#glm residuals
simulationOutput <- simulateResiduals(fittedModel = finalHead, n=999)
plotSimulatedResiduals(simulationOutput = simulationOutput)
```



```
respV <- "Saliva_mean"
predVs <- c("Temp_mean_mean","RH_mean_mean", "DTR_mean", "Temp_min_mean", "RH_min_mean", "Temp_max_mean

myCols <- c(respV, predVs, "block","class", "site")
modDF <- seasonInfSite %>%
  dplyr::select(one_of(myCols))
modDF <- na.omit(modDF)

#model selection for initial variable (bc of high correlation)
m1 <- glm(Saliva_mean ~ Temp_mean_mean,
          data=modDF,
          family=gaussian)
m2 <- glm(Saliva_mean ~ RH_mean_mean,
          data=modDF,
```

```r
          family=gaussian)
m3 <- glm(Saliva_mean ~ DTR_mean,
          data=modDF,
          family=gaussian)
m4 <- glm(Saliva_mean ~ Temp_min_mean,
          data=modDF,
          family=gaussian)
m5 <- glm(Saliva_mean ~ Temp_max_mean,
          data=modDF,
          family=gaussian)
m6 <- glm(Saliva_mean ~ RH_min_mean,
          data=modDF,
          family=gaussian)
m7 <- glm(Saliva_mean ~ RH_max_mean,
          data=modDF,
          family=gaussian)

modelSums <- do.call(rbind, lapply(list(m1,m2,m3, m4, m5, m6, m7), broom::glance)) #2 seems best, but t
tidy(m2)
```

```
##           term   estimate   std.error statistic    p.value
## 1  (Intercept) -0.2856187 0.222758502 -1.282190 0.21804309
## 2 RH_mean_mean  0.0045945 0.002597975  1.768493 0.09603649
```

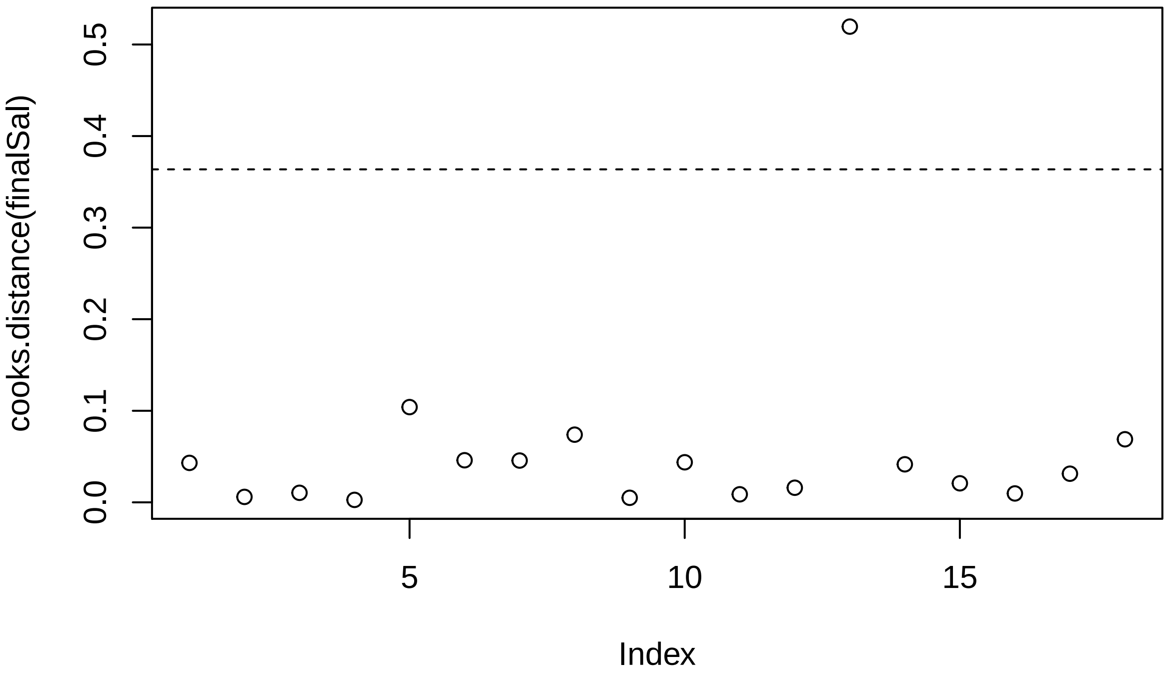
```r
#model selection
m1 <- glm(Saliva_mean ~ Temp_mean_mean,
          data=modDF,
          family=gaussian)
m2 <- glm(Saliva_mean ~ RH_mean_mean,
          data=modDF,
          family=gaussian)

m3 <- glm(Saliva_mean ~ Temp_mean_mean+RH_mean_mean,
          data=modDF,
          family=gaussian)
m4 <- glm(Saliva_mean ~ Temp_mean_mean*RH_mean_mean,
          data=modDF,
          family=gaussian)

modelSums <- do.call(rbind, lapply(list(m1,m2, m3, m4), broom::glance)) #2 seems best

AICc(m1,m2, m3, m4)
```

```
##    df      AICc
## m1  3 -29.49212
## m2  3 -31.70849
## m3  4 -28.52592
## m4  5 -24.73537
```

```r
tidy(m2)
```

```
##           term   estimate   std.error statistic    p.value
## 1  (Intercept) -0.2856187 0.222758502 -1.282190 0.21804309
## 2 RH_mean_mean  0.0045945 0.002597975  1.768493 0.09603649
```

```
finalSal <- glm(Saliva_mean ~ RH_mean_mean,
                data=modDF, #drop outlier
                family=gaussian)

plot(cooks.distance(finalSal))
abline(h=4/length(modDF), lty=2)
```



```
car::Anova(finalSal)
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: Saliva_mean
##              LR Chisq Df Pr(>Chisq)
## RH_mean_mean   3.1276  1    0.07698 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#confint(finalSal)
drop1(finalSal, test="F")
```

```
## Single term deletions
##
## Model:
## Saliva_mean ~ RH_mean_mean
##              Df Deviance     AIC F value  Pr(>F)
## <none>            0.11793 -33.423
## RH_mean_mean  1   0.14098 -32.209  3.1276 0.09604 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

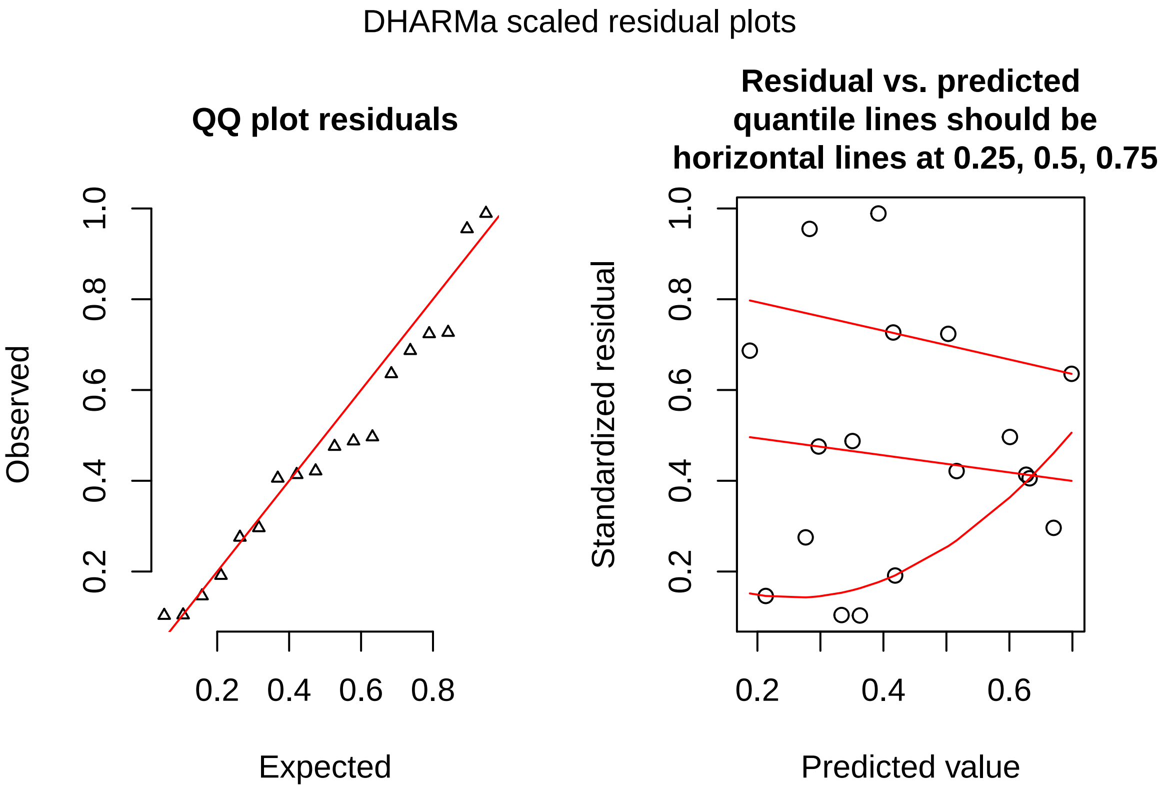
```
tidy(finalSal)
```

```
##           term   estimate   std.error statistic    p.value
## 1  (Intercept) -0.2856187 0.222758502 -1.282190 0.21804309
## 2 RH_mean_mean  0.0045945 0.002597975  1.768493 0.09603649
```

```
outlierTest(finalSal)
```

```
##    rstudent unadjusted p-value Bonferonni p
## 13 3.186174          0.0014417      0.02595
```

```
#glm residuals
simulationOutput <- simulateResiduals(fittedModel = finalHead, n=999)
plotSimulatedResiduals(simulationOutput = simulationOutput)
```



Infection x Temperature Plot

```
#pdf(file="figures/forMS/InfxTemp3Panel.pdf", width = 8, height=3, family="sans")
plotBody <- ggplot(data=seasonInfSite, aes(x=Temp_mean_mean, y=Body_mean))+
  geom_errorbar(aes(ymin=Body_mean-Body_se, ymax=Body_mean+Body_se), color=errorColor, width=0.1) +
  geom_point(aes(color=class, shape=block), size=4.5) +
  scale_color_manual(values=c(colR, colS, colU)) +
  theme_fivethirtyeight() +
  theme(panel.background = element_rect(fill = "transparent", colour = NA),
        plot.background = element_rect(fill = "transparent", colour = NA),
        legend.key = element_blank(),
        legend.background = element_rect(fill = "transparent", colour = NA),
        axis.title = element_text(),
        axis.title.x = element_text(),
        axis.title.y=element_text(),
        axis.line=element_line(color=axisColor, size=0.5),
        panel.grid = element_blank())+
  ylab("Prop. Infected")+
  xlab("") +
  theme(legend.position="none")
```

```
plotHead <- ggplot(data=seasonInfSite, aes(x=Temp_mean_mean, y=Head_mean))+
   geom_errorbar(aes(ymin=Head_mean-Head_se, ymax=Head_mean+Head_se), color=errorColor, width=0.1) +
  geom_point(aes(color=class, shape=block), size=4) +
  scale_color_manual(values=c(colR, colS, colU)) +
  theme_fivethirtyeight() +
    theme(panel.background = element_rect(fill = "transparent", colour = NA),
        plot.background = element_rect(fill = "transparent", colour = NA),
        legend.key = element_blank(),
        legend.background = element_rect(fill = "transparent", colour = NA),
        axis.title = element_text(),
        axis.title.x = element_text(),
        axis.title.y=element_blank(),
        axis.line=element_line(color=axisColor, size=0.5),
        panel.grid = element_blank())+
  #ylab("Prop. Disseminated")+
  xlab("Mean Daily Temperature (C)")+
  theme(legend.position="none")

plotSaliva <- ggplot(data=seasonInfSite, aes(x=Temp_mean_mean, y=Saliva_mean))+
  geom_errorbar(aes(ymin=Saliva_mean-Saliva_se, ymax=Saliva_mean+Saliva_se), color=errorColor, width=0.
  #geom_errorbarh(aes(xmin=Temp_mean_mean-Temp_mean_se, xmax=Temp_mean_mean+Temp_mean_se, y=Body_mean),
  geom_point(aes(color=class, shape=block), size=4.5) +
  scale_color_manual(values=c(colR, colS, colU)) +
  theme_fivethirtyeight() +
    theme(panel.background = element_rect(fill = "transparent", colour = NA),
        plot.background = element_rect(fill = "transparent", colour = NA),
        legend.key = element_blank(),
        legend.background = element_rect(fill = "transparent", colour = NA),
        axis.title = element_text(),
        axis.title.x = element_text(),
        axis.title.y=element_blank(),
        axis.line=element_line(color=axisColor, size=0.5),
        panel.grid = element_blank()) +
  #ylab("Prop. Infectious")+
  xlab("") +
  scale_y_continuous(limits=c(0,1))+
  #theme(legend.position="right", legend.direction="vertical")
  theme(legend.position="none")


plot_grid(plotBody, plotHead, plotSaliva,
          labels=c("A", "B", "C"),
          ncol=3)
#dev.off()
```

Infection x Relative Humidity Plot

```
#png(file="figures/forMS/supplement/InfxRH3Panel.png", width = 8, height=3, units="in", res=500, family
plotBody <- ggplot(data=seasonInfSite, aes(x=RH_mean_mean, y=Body_mean))+
  geom_errorbar(aes(ymin=Body_mean-Body_se, ymax=Body_mean+Body_se), color=errorColor, width=0.1) +
  geom_point(aes(color=class, shape=block), size=4.5) +
  scale_color_manual(values=c(colR, colS, colU)) +
  theme_fivethirtyeight() +
  theme(panel.background = element_rect(fill = "transparent", colour = NA),
        plot.background = element_rect(fill = "transparent", colour = NA),
        legend.key = element_blank(),
        legend.background = element_rect(fill = "transparent", colour = NA),
        axis.title = element_text(),
        axis.title.x = element_text(),
        axis.title.y=element_text(),
        axis.line=element_line(color=axisColor, size=0.5),
        panel.grid = element_blank())+
  ylab("Prop. Infected")+
  xlab("") +
  theme(legend.position="none")

plotHead <- ggplot(data=seasonInfSite, aes(x=RH_mean_mean, y=Head_mean))+
   geom_errorbar(aes(ymin=Head_mean-Head_se, ymax=Head_mean+Head_se), color=errorColor, width=0.1) +
  geom_point(aes(color=class, shape=block), size=4) +
  scale_color_manual(values=c(colR, colS, colU)) +
  theme_fivethirtyeight() +
    theme(panel.background = element_rect(fill = "transparent", colour = NA),
        plot.background = element_rect(fill = "transparent", colour = NA),
        legend.key = element_blank(),
        legend.background = element_rect(fill = "transparent", colour = NA),
        axis.title = element_text(),
        axis.title.x = element_text(),
        axis.title.y=element_blank(),
        axis.line=element_line(color=axisColor, size=0.5),
        panel.grid = element_blank())+
  #ylab("Prop. Disseminated")+
  xlab("Relative Humidity (%)")+
  theme(legend.position="none")

plotSaliva <- ggplot(data=seasonInfSite, aes(x=RH_mean_mean, y=Saliva_mean))+
  geom_errorbar(aes(ymin=Saliva_mean-Saliva_se, ymax=Saliva_mean+Saliva_se), color=errorColor, width=0.
  geom_point(aes(color=class, shape=block), size=4.5) +
  scale_color_manual(values=c(colR, colS, colU)) +
  theme_fivethirtyeight() +
    theme(panel.background = element_rect(fill = "transparent", colour = NA),
        plot.background = element_rect(fill = "transparent", colour = NA),
        legend.key = element_blank(),
        legend.background = element_rect(fill = "transparent", colour = NA),
        axis.title = element_text(),
        axis.title.x = element_text(),
        axis.title.y=element_blank(),
        axis.line=element_line(color=axisColor, size=0.5),
        panel.grid = element_blank()) +
  #ylab("Prop. Infectious")+
  xlab("") +
  scale_y_continuous(limits=c(0,1))+
  #theme(legend.position="right", legend.direction="vertical")
  theme(legend.position="none")


plot_grid(plotBody, plotHead, plotSaliva,
          labels=c("A", "B", "C"),
          ncol=3)
#dev.off()
```

Infection and Body Size

Body Infection

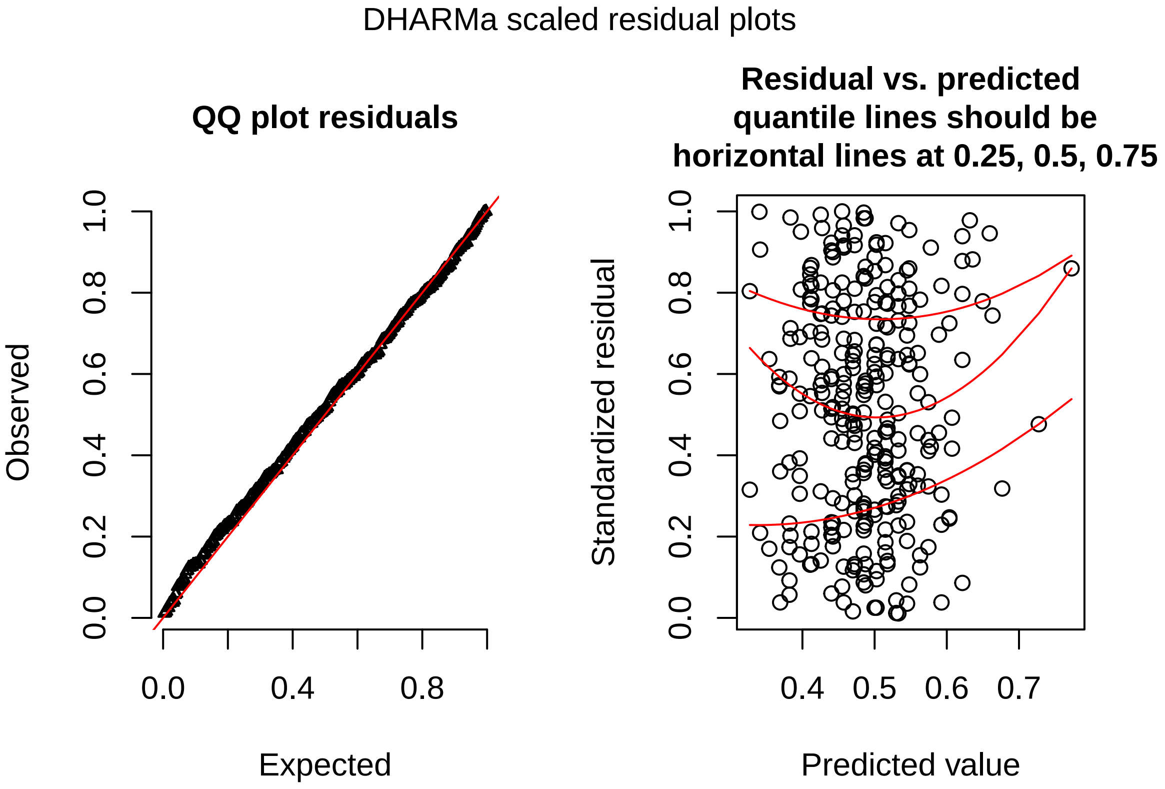
```
bodyWingDF <- seasonInf %>%
  dplyr::select(Body, Wing, site, block, class) %>%
  filter(!is.na(Body)) %>%
  filter(!is.na(Wing))


bodyWing <- glmer(Body~Wing + (1|site),
                          data=bodyWingDF,
                          family=binomial(link="logit"))

tidy(bodyWing)
```

```
##                  term   estimate std.error statistic    p.value group
## 1         (Intercept)  4.9878964 2.3128770  2.156577 0.03103867 fixed
## 2                Wing -2.0232763 0.9260367 -2.184877 0.02889787 fixed
## 3 sd_(Intercept).site  0.2128793        NA        NA         NA  site
```

```
summary(bodyWing)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Body ~ Wing + (1 | site)
##    Data: bodyWingDF
##
##      AIC      BIC   logLik deviance df.resid
##    399.5    410.5   -196.8    393.5      286
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.5784 -0.9612 -0.6971  0.9858  1.5234
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  site   (Intercept) 0.04532  0.2129
## Number of obs: 289, groups:  site, 9
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)    4.988      2.313   2.157   0.0310 *
## Wing          -2.023      0.926  -2.185   0.0289 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##      (Intr)
## Wing -0.998
```

```
simulationOutput <- simulateResiduals(fittedModel = bodyWing, n=999)
plotSimulatedResiduals(simulationOutput = simulationOutput)
```



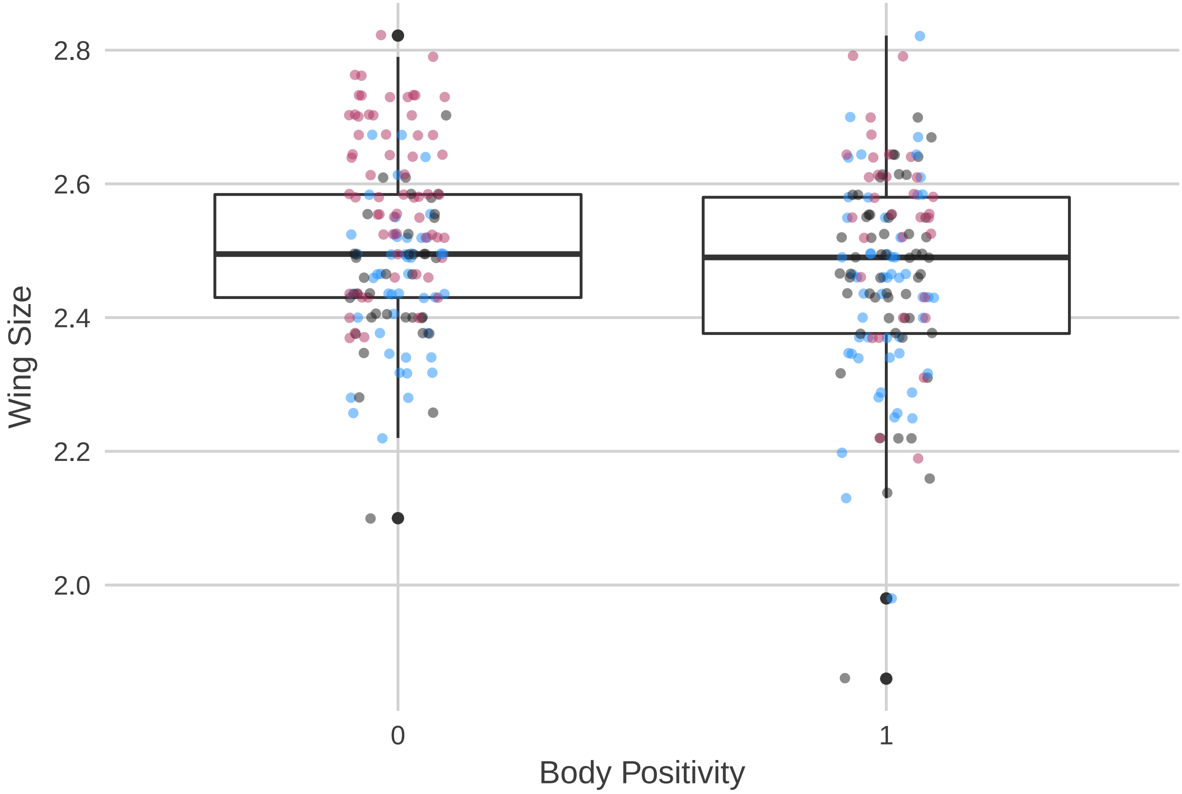
```
Anova(bodyWing)
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: Body
##       Chisq Df Pr(>Chisq)
## Wing 4.7737  1     0.0289 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
drop1(bodyWing, test="Chisq")
```

```
## Single term deletions
##
## Model:
## Body ~ Wing + (1 | site)
##        Df    AIC    LRT Pr(Chi)
## <none>    399.51
## Wing    1 402.21 4.6937 0.03027 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
ggplot(data=bodyWingDF, aes(x=factor(Body), y=Wing))+
  geom_boxplot() +
  geom_jitter(shape=16, position=position_jitter(0.1), aes(color=factor(class)), alpha=0.5) +
    theme_fivethirtyeight()+
  theme(axis.title = element_text()) +
  theme(panel.background = element_rect(fill = "transparent", colour = NA),
        plot.background = element_rect(fill = "transparent", colour = NA),
        legend.key = element_blank())+
  theme(legend.background = element_rect(fill = "transparent", colour = NA))+
  theme(legend.position="right", legend.direction="vertical") +
  scale_color_manual(values=c("dodgerblue", "gray10", "maroon"), labels=c("Rural", "Suburban", "Urban")
    scale_fill_manual(values=c("dodgerblue", "gray80", "maroon"), labels=c("Rural", "Suburban", "Urban"
  guides(color=F) +
  xlab("Body Positivity") +
  ylab("Wing Size")
```

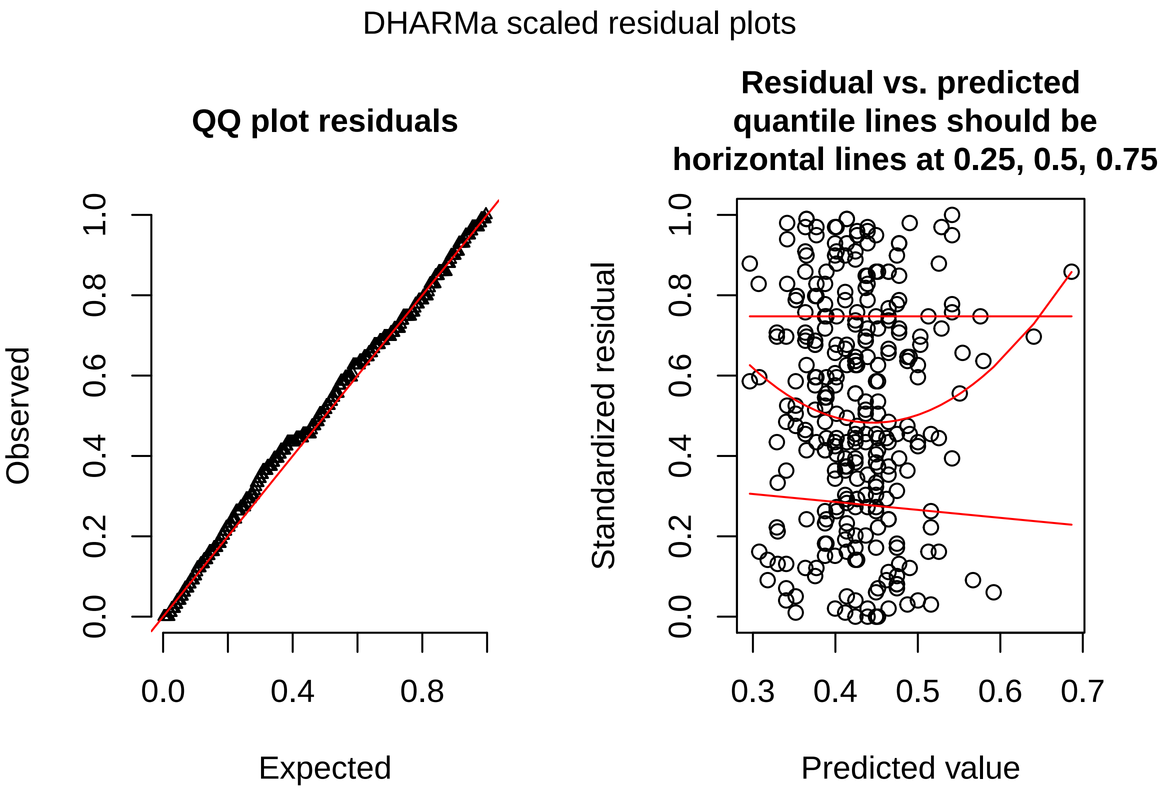


Head Infection

```
headWingDF <- seasonInf %>%
  dplyr::select(Head, Wing, site, block, class) %>%
  filter(!is.na(Head)) %>%
  filter(!is.na(Wing))


headWing <- glmer(Head~Wing + (1|site),
                          data=headWingDF,
                          family=binomial(link="logit"))
# headWing <- lmer(Wing~Head + (1|site),
#                            data=headWingDF) #this tests if the wing length is significantly different
tidy(headWing)
```

```
##                  term   estimate std.error statistic    p.value group
## 1         (Intercept)  3.9718215 2.3146765  1.715929 0.08617496 fixed
## 2                Wing -1.7142078 0.9275247 -1.848153 0.06458016 fixed
## 3 sd_(Intercept).site  0.1861565        NA        NA         NA  site
```

```
simulationOutput <- simulateResiduals(fittedModel = headWing, n=99)
plotSimulatedResiduals(simulationOutput = simulationOutput)
```



```
Anova(headWing)
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: Head
##       Chisq Df Pr(>Chisq)
## Wing 3.4157  1    0.06458 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
drop1(headWing, test="Chisq")
```

```
## Single term deletions
##
## Model:
## Head ~ Wing + (1 | site)
##        Df    AIC    LRT Pr(Chi)
## <none>    396.88
## Wing    1 398.18 3.3029 0.06916 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Saliva Infection

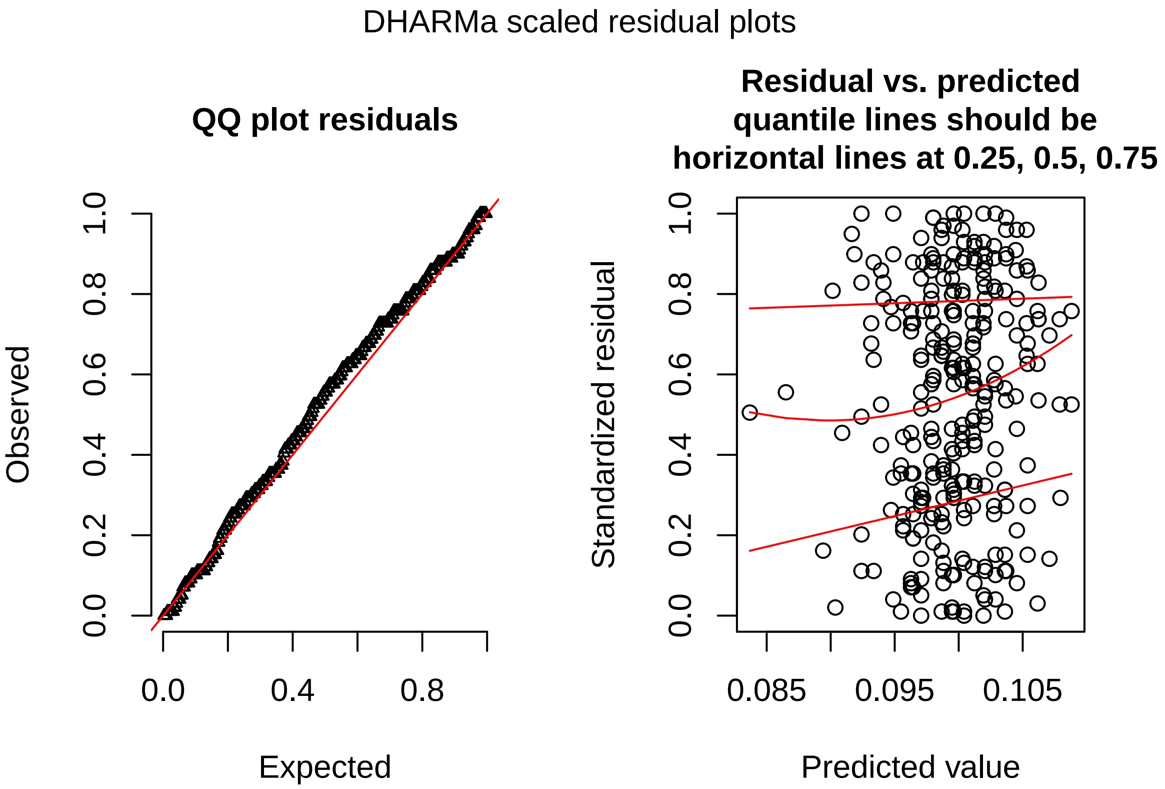
```
salWingDF <- seasonInf %>%
  dplyr::select(Saliva, Wing, site, block, class) %>%
  filter(!is.na(Saliva)) %>%
  filter(!is.na(Wing))


salWing <- glmer(Saliva~Wing + (1|site),
                          data=salWingDF,
                          family=binomial(link="logit"))
summary(salWing)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Saliva ~ Wing + (1 | site)
##    Data: salWingDF
##
##      AIC      BIC   logLik deviance df.resid
##    194.7    205.7    -94.4    188.7      288
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -0.3494 -0.3369 -0.3324 -0.3267  3.1482
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  site   (Intercept) 0        0
## Number of obs: 291, groups:  site, 9
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -2.9552     3.5124  -0.841     0.40
## Wing          0.3021     1.4033   0.215     0.83
##
## Correlation of Fixed Effects:
##      (Intr)
## Wing -0.998
```

```
tidy(salWing)
```

```
##                  term   estimate std.error  statistic   p.value group
## 1         (Intercept) -2.9551864  3.512354 -0.8413692 0.4001411 fixed
## 2                Wing  0.3020559  1.403287  0.2152488 0.8295734 fixed
## 3 sd_(Intercept).site  0.0000000        NA         NA        NA  site
```

```
simulationOutput <- simulateResiduals(fittedModel = salWing, n=99)
plotSimulatedResiduals(simulationOutput = simulationOutput)
```



```r
Anova(salWing)
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: Saliva
##       Chisq Df Pr(>Chisq)
## Wing 0.0463  1     0.8296
```

```r
drop1(salWing, test="Chisq")
```

```
## Single term deletions
##
## Model:
## Saliva ~ Wing + (1 | site)
##        Df    AIC      LRT Pr(Chi)
## <none>    194.71
## Wing    1 192.76 0.046633   0.829
```

Direct Effects

Data Format and Loading

```r
emergAug <- read.csv("../data/emergence/raw/AugustEmergence.csv")
emergAug$block <- "summer"

emergOct <- read.csv("../data/emergence/raw/OctoberEmergence.csv")
emergOct$block <- "fall"
```

```r
#drop U3T1 in fall because it was eaten by ants
emergOct <- filter(emergOct, Tray_Code!="U3T1")
emergOct <- filter(emergOct, Tray_Code!="S1T3") #dumped
emergOct <- filter(emergOct, Tray_Code!="U1T4") #dumped

emergAll <- rbind(emergOct, emergAug)
#expand so each mosquito gets one row
emergExp <- emergAll[rep(seq.int(1,nrow(emergAll)), emergAll$Num_Emerge),
                     c(11, 2,4,5,6:9)]
#sum(emergAll$Num_Emerge)==nrow(emergExp) #quick check this worked
emergTray <- emergExp %>%
  filter(Sex=="F") %>%
  mutate(devRate=1/Exp_Day) %>%
  group_by(Tray_Code, Site_Code, Class, block) %>%
  summarise(devRate=mean(devRate, na.rm=T)) %>%
  ungroup() %>%
  mutate(block=case_when(
    block=="summer" ~ "Summer",
    block=="fall"   ~ "Fall"
  ))
emergTray$block <- factor(emergTray$block, levels=c("Summer", "Fall"))

#get survival per tray
survSumm <- emergExp %>%
  filter(Sex=="F") %>%
  group_by(block, Tray_Code) %>%
  dplyr::mutate(percSurv=n()) %>%
  ungroup() %>%
  dplyr::select(block, Class, Site_Code, Tray_Code, percSurv)
survSumm <- unique(survSumm)
```

```r
fillIn <- function(df, endDay, totalMosq=50){
  #' Fill In Emergence Dates
  #' this function fills in for those mosquitoes that did not emerge so we do not have data for, it giv
  #' @param df the data frame you wish to fill in, in our case by pot
  #' @param endDay the last day of emergence
  #' @param totalMosq estimated starting number of mosquitoes per pot
  #' @returns dataframe with census data filled in for mosquitoes that did not emerge
  toRep <- df[1,]
  toRep$Exp_Day <- endDay
  toRep$event <- 0
  if(nrow(df)<totalMosq){
    toAdd <- toRep[rep(1, (totalMosq-nrow(df))),]
    allTest <- rbind(df, toAdd)
  } else {
    toAdd <- NA
    allTest <- NA
  }
  return(allTest)
}
```

```r
applyFill <- function(season, allData=emergExp){
  #' Apply FillIn function
  #' @param season "fall" or "summer"
  #' @param allData full dataframe with row for each mosquito that emerged
  #' @returns censused data for the full season
  tempList <- list()
  tempDF <- allData
  tempDF <- tempDF[tempDF$Sex=="F",]
  tempDF <- tempDF[tempDF$block==season,]
  tempDF$event <- 1 #add emergence event
  for (i in 1:length(unique(tempDF$Tray_Code))){
    df <- tempDF[tempDF$Tray_Code==unique(tempDF$Tray_Code)[i],]
    endDay <- max(tempDF$Exp_Day)
    tempList[[i]] <- fillIn(df=df, endDay=endDay)
  }
  allSurv <- do.call(rbind.data.frame, tempList)
  return(allSurv)
}
```

```
summerSurv <- applyFill(season="summer")
fallSurv <- applyFill(season="fall")
allSurv <- rbind(summerSurv, fallSurv)

survTray <- allSurv %>%
  filter(Sex=="F") %>%
  group_by(Tray_Code, Site_Code, Class, block) %>%
  summarise(survival=mean(event))
survTray$block[survTray$block=="summer"] <- "Summer"
survTray$block[survTray$block=="fall"] <- "Fall"
survTray$block <- factor(survTray$block, levels=c("Summer", "Fall"))
```

Tie in tray-level climate data:

```
allClim <- rbind(augClim, octClim)
allClim$block <- factor(rep(c("Summer", "Fall"), each=34), levels=c("Summer", "Fall"))
survClim <- merge(survTray, allClim, by.x=c("block","Tray_Code"), by.y=c("block","Tray_ID"))

emergClim <- merge(emergTray, allClim, by.x=c("block","Tray_Code"), by.y=c("block","Tray_ID"))
```

Survival

Class x Season

```
#model selection by AIC and logLik
m1 <- glmer(survival ~ block + (1|Site_Code),
                   data=survTray,
                 family=gaussian(link="logit"))

m2 <- glmer(survival ~ Class + (1|Site_Code),
                   data=survTray,
                 family=gaussian(link="logit"))

m3 <- glmer(survival ~ Class + block + (1|Site_Code),
                   data=survTray,
                 family=gaussian(link="logit"))
```

```
m4 <- glmer(survival ~ Class * block + (1|Site_Code),
                   data=survTray,
                 family=gaussian(link="logit"))

modelSums <- do.call(rbind, lapply(list(m1,m2,m3, m4), broom::glance))
modelSums #m1 is best, but near m3 & m4
```

```
##       sigma    logLik       AIC       BIC deviance df.residual
## 1 0.1658887 23.119655 -38.23931 -29.42054 1.693916          63
## 2 0.2422129 -1.621092  13.24218  24.26565 3.848056          62
## 3 0.1658848 23.140524 -34.28105 -21.05289 1.693094          61
## 4 0.1593213 25.878542 -35.75708 -18.11954 1.558505          59
```

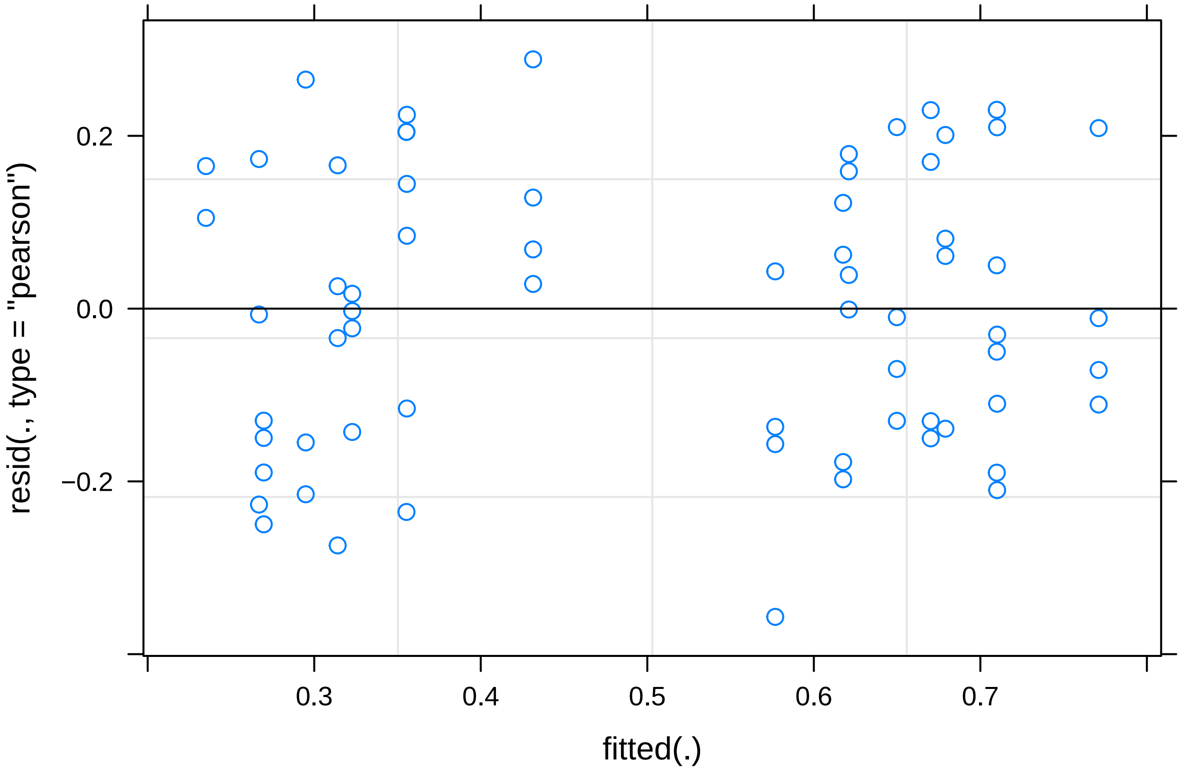
```
AICc(m1,m2,m3, m4) #still m1
```

```
##    df      AICc
## m1  4 -37.59415
## m2  5  14.22579
## m3  6 -32.88105
## m4  8 -33.27433
```

```
SurvModSeason <- glmer(survival ~ block + (1|Site_Code),
                   data=survTray,
                 family=gaussian(link="logit"))

summary(SurvModSeason)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: gaussian  ( logit )
## Formula: survival ~ block + (1 | Site_Code)
##    Data: survTray
##
##      AIC      BIC   logLik deviance df.resid
##    -38.2    -29.4     23.1    -46.2       63
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.15106 -0.83144 -0.01688  0.91427  1.73948
##
## Random effects:
##  Groups    Name        Variance Std.Dev.
##  Site_Code (Intercept) 0.10886  0.3299
##  Residual              0.02752  0.1659
## Number of obs: 67, groups:  Site_Code, 9
##
## Fixed effects:
##             Estimate Std. Error t value Pr(>|z|)
## (Intercept)   0.7042     0.2288   3.078  0.00208 **
## blockFall    -1.4902     0.1917  -7.775 7.55e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##           (Intr)
## blockFall -0.381
```

```
modResults <- tidy(SurvModSeason)
plot(SurvModSeason)
```



```
car::Anova(SurvModSeason)
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: survival
##       Chisq Df Pr(>Chisq)
## block 60.45  1  7.549e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#pairwise
survPair <- pairs(lsmeans::lsmeans(SurvModSeason, ~block))
survPair
```

```
##  contrast      estimate        SE df z.ratio p.value
##  Summer - Fall 1.490186 0.1916654 NA   7.775  <.0001
##
## Results are given on the log odds ratio (not the response) scale.
```

```
#coefficients
boot::inv.logit(modResults$estimate[1]) #fall
```

```
## [1] 0.6691121
```

```
boot::inv.logit(modResults$estimate[1]+modResults$estimate[2]) #summer
```

```
## [1] 0.3130252
```

Microclimate

Model selection

```r
respV <- "survival"
predVs <- c("Temp_mean","RH_mean", "DTR", "Temp_min", "RH_min", "Temp_max", "RH_max")

myCols <- c(respV, predVs, "block","Class", "Site_Code")
modDF <- survClim %>%
  dplyr::select(one_of(myCols))
modDF <- na.omit(modDF)

#model selection for initial variable

m1 <- lmer(survival~Temp_mean + (1|Site_Code),
           data=modDF)

m2 <- lmer(survival~RH_mean + (1|Site_Code),
           data=modDF)

m3 <- lmer(survival~DTR + (1|Site_Code),
           data=modDF)

m4 <- lmer(survival~Temp_min + (1|Site_Code),
           data=modDF)

m5 <- lmer(survival~RH_min + (1|Site_Code),
           data=modDF)

m6 <- lmer(survival~Temp_max + (1|Site_Code),
           data=modDF)

m7 <- lmer(survival~RH_max + (1|Site_Code),
           data=modDF)

modelSums <- do.call(rbind, lapply(list(m1,m2,m3, m4, m5, m6, m7), broom::glance))
modelSums #m1 or m4
```

```
##       sigma    logLik        AIC        BIC   deviance df.residual
## 1 0.1621684 15.571069 -23.142137 -14.569598 -44.577817          59
## 2 0.1837171  4.483259  -0.966518   7.606021 -22.715586          59
## 3 0.1775250 10.431923 -12.863847  -4.291308 -33.640148          59
## 4 0.1627902 15.193627 -22.387254 -13.814715 -44.485120          59
## 5 0.1794725  7.550733  -7.101467   1.471072 -30.579517          59
## 6 0.1864933  7.270278  -6.540555   2.031984 -26.644909          59
## 7 0.2517561 -8.273081  24.546163  33.118702   4.570757          59
```

```r
Weights(AICc(m1,m2,m3, m4, m5, m6, m7))
```

```
## [1] 5.909927e-01 9.040816e-06 3.464812e-03 4.051924e-01 1.942651e-04
## [6] 1.467555e-04 2.607350e-11
```

```r
tidy(m1)
```

```
##                        term    estimate  std.error statistic     group
## 1               (Intercept) -0.74564962 0.145368674 -5.129369     fixed
## 2                Temp_mean  0.05430927 0.006132529  8.855935     fixed
## 3 sd_(Intercept).Site_Code  0.07289187          NA        NA Site_Code
## 4  sd_Observation.Residual  0.16216844          NA        NA  Residual
```

```
#find covariates that aren't correlated
covars <- data.frame(cor(survClim[,6:12]))
rownames(covars[abs(covars$Temp_mean)<0.8,]) #only RH_mean and RH_max
```

```
## [1] "RH_mean" "RH_max"
```

```
#model selection
m1 <- lmer(survival~Temp_mean + (1|Site_Code),
           data=modDF)

m2 <- lmer(survival~Temp_mean + RH_mean + (1|Site_Code),
           data=modDF)

m3 <- lmer(survival~Temp_mean * RH_mean + (1|Site_Code),
           data=modDF)

m4 <- lmer(survival~RH_mean + (1|Site_Code),
           data=modDF)

modelSums <- do.call(rbind, lapply(list(m1,m2, m3, m4), broom::glance)) #1 seems best
modelSums
```

```
##       sigma    logLik         AIC        BIC  deviance df.residual
## 1 0.1621684 15.571069 -23.1421372 -14.569598 -44.57782          59
## 2 0.1627397 11.266002 -12.5320031  -1.816329 -44.61667          58
## 3 0.1641516  6.089038  -0.1780767  12.680732 -44.92508          57
## 4 0.1837171  4.483259  -0.9665180   7.606021 -22.71559          59
```

```
AICc(m1,m2, m3, m4)
```

```
##    df        AICc
## m1  4 -22.4524820
## m2  5 -11.4793715
## m3  6   1.3219233
## m4  4  -0.2768628
```

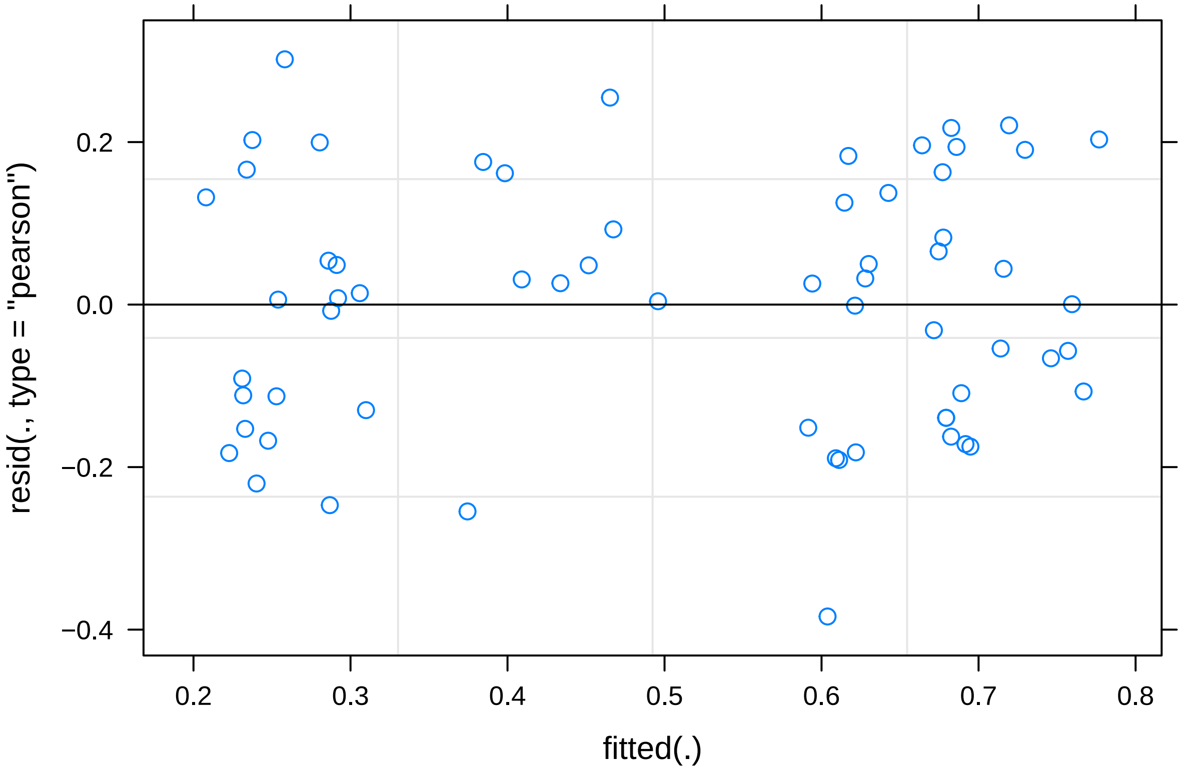
```
tidy(m1)
```

```
##                       term    estimate   std.error statistic     group
## 1              (Intercept) -0.74564962 0.145368674 -5.129369     fixed
## 2                Temp_mean  0.05430927 0.006132529  8.855935     fixed
## 3 sd_(Intercept).Site_Code  0.07289187          NA        NA Site_Code
## 4  sd_Observation.Residual  0.16216844          NA        NA  Residual
```

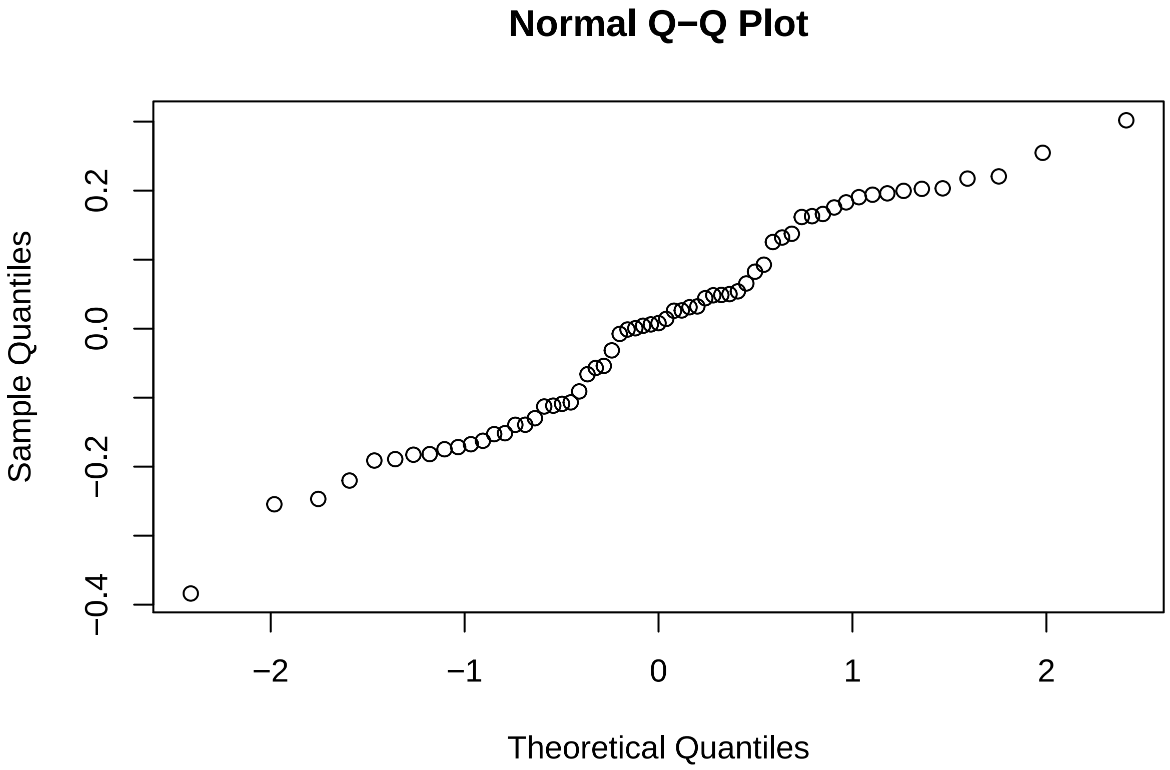
```
survModClim <- glmer(survival~Temp_mean+(1|Site_Code),
                     data=survClim,
                     family=gaussian("logit"))

summary(survModClim)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: gaussian  ( logit )
## Formula: survival ~ Temp_mean + (1 | Site_Code)
##    Data: survClim
##
##      AIC      BIC   logLik deviance df.resid
##    -40.2    -31.6     24.1    -48.2       59
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.40767 -0.84428  0.04972  0.84500  1.89339
##
## Random effects:
##  Groups    Name        Variance Std.Dev.
##  Site_Code (Intercept) 0.11266  0.3357
##  Residual              0.02542  0.1594
## Number of obs: 63, groups:  Site_Code, 9
##
## Fixed effects:
##             Estimate Std. Error t value Pr(>|z|)
## (Intercept) -5.53248    0.73027  -7.576 3.57e-14 ***
## Temp_mean    0.23995    0.02966   8.089 6.02e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##           (Intr)
## Temp_mean -0.956
```

```r
modResults <- tidy(survModClim)

plot(survModClim)
```



```
qqnorm(resid(survModClim))
```



```
#coefficients
modResults$estimate[2] #temp coefficient
```

```
## [1] 0.2399536
```

Emergence

Class x Season

Model Selection

```
#model selection
m0 <- lmer(devRate ~ 1 + (1|Site_Code),
           data=emergTray)
m1 <- lmer(devRate ~ Class +(1|Site_Code),
                   data=emergTray)

m2 <- lmer(devRate ~ block +(1|Site_Code),
                   data=emergTray)

m3 <- lmer(devRate ~ Class * block +(1|Site_Code),
                   data=emergTray)

m4 <- lmer(devRate ~ Class + block +(1|Site_Code),
                   data=emergTray)

modelSums <- do.call(rbind, lapply(list(m0, m1,m2,m3, m4), broom::glance))
modelSums #m2 is best
```

```
##         sigma   logLik       AIC       BIC  deviance df.residual
## 1 0.019849282 162.9405 -319.8810 -313.2669 -336.0945          64
## 2 0.019767728 155.6473 -301.2945 -290.2711 -338.7079          62
## 3 0.006157888 228.8366 -449.6732 -440.8544 -479.8224          63
## 4 0.006095069 213.1806 -410.3611 -392.7236 -487.0683          59
## 5 0.006160639 221.1502 -430.3004 -417.0723 -483.8547          61
```

```
AICc(m0, m1,m2,m3, m4) #still m2
```

```
##    df      AICc
## m0  3 -319.5000
## m1  5 -300.3109
## m2  4 -449.0280
## m3  8 -407.8784
## m4  6 -428.9004
```

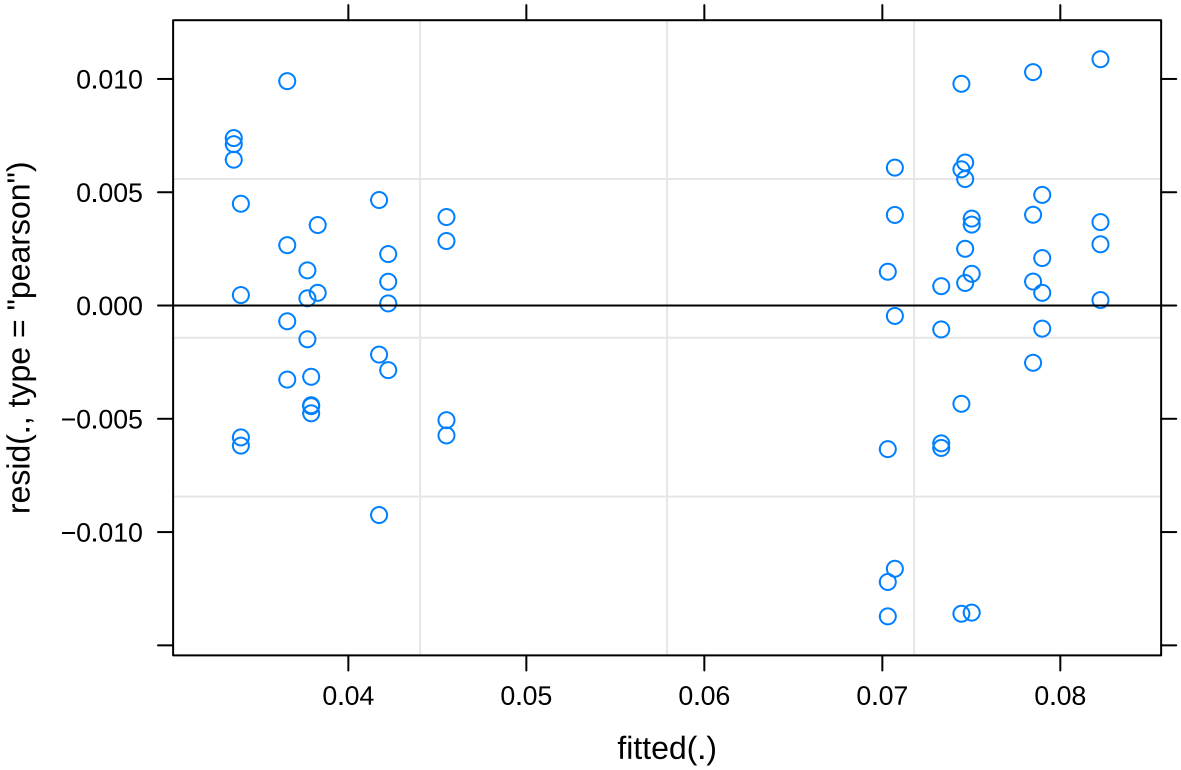
```
mixEmergeTray <- lmer(devRate ~ block +(1|Site_Code),
                   data=emergTray)

summary(mixEmergeTray)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: devRate ~ block + (1 | Site_Code)
##    Data: emergTray
##
## REML criterion at convergence: -457.7
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.2281 -0.6178  0.1385  0.6286  1.7654
##
## Random effects:
##  Groups    Name        Variance  Std.Dev.
##  Site_Code (Intercept) 1.947e-05 0.004412
##  Residual              3.792e-05 0.006158
## Number of obs: 67, groups:  Site_Code, 9
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)  0.075350   0.001793   42.02
## blockFall   -0.036743   0.001515  -24.25
##
## Correlation of Fixed Effects:
##           (Intr)
## blockFall -0.388
```

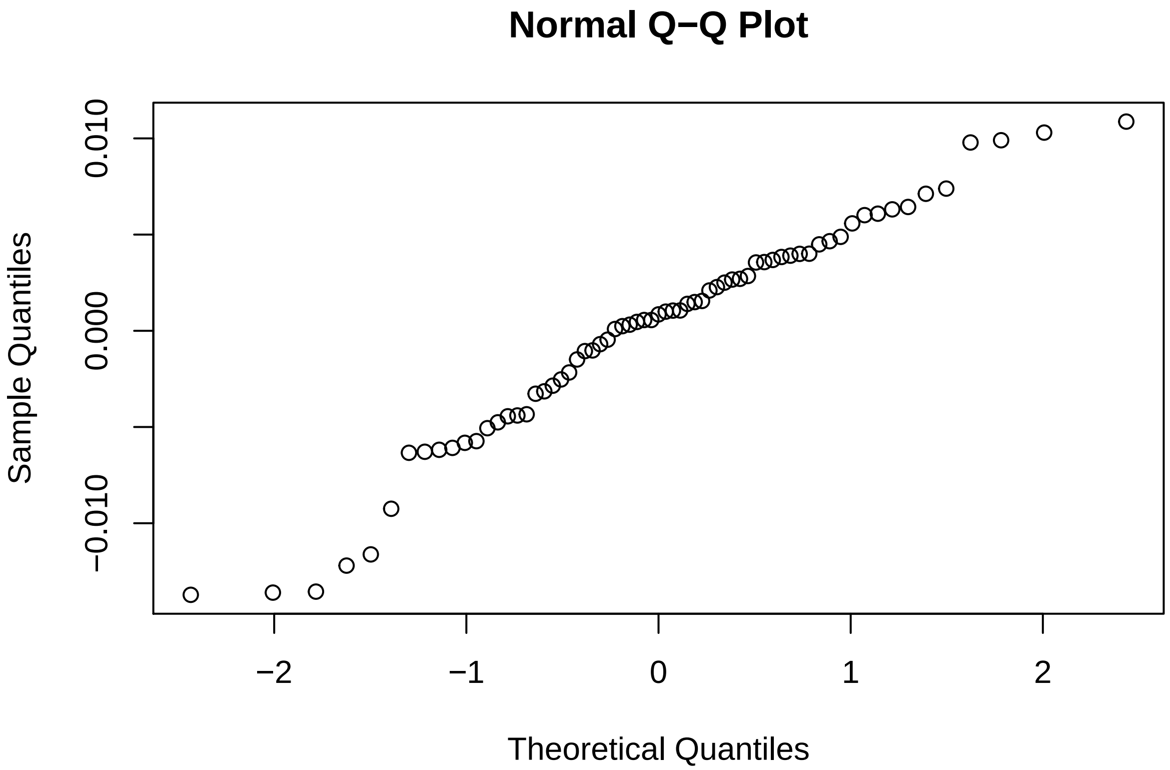
```
tidy(mixEmergeTray)
```

```
##                       term     estimate   std.error statistic     group
## 1              (Intercept)  0.075350056 0.001793406  42.01506     fixed
## 2                blockFall -0.036742816 0.001515196 -24.24955     fixed
## 3 sd_(Intercept).Site_Code  0.004412125          NA        NA Site_Code
## 4  sd_Observation.Residual  0.006157888          NA        NA  Residual
```

```
plot(mixEmergeTray)
```



```
qqnorm(resid(mixEmergeTray))
```



```
#confint(mixEmerge)
Anova(mixEmergeTray)
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: devRate
##        Chisq Df Pr(>Chisq)
## block 588.04  1  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#pairwise
summary(multcomp::glht(mixEmergeTray, linfct = multcomp::mcp(block = "Tukey"), test = adjusted("holm"))
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: lmer(formula = devRate ~ block + (1 | Site_Code), data = emergTray)
##
## Linear Hypotheses:
##                       Estimate Std. Error z value Pr(>|z|)
## Fall - Summer == 0 -0.036743   0.001515  -24.25   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

Microclimate

Model Selection:

```
respV <- "devRate"
predVs <- c("Temp_mean","RH_mean", "DTR", "Temp_min", "RH_min", "Temp_max", "RH_max")

myCols <- c(respV, predVs, "block","Class", "Site_Code")
modDF <- emergClim %>%
  dplyr::select(one_of(myCols))
modDF <- na.omit(modDF)

#model selection for initial variable
m0 <- lmer(devRate~1 + (1|Site_Code),
           data=modDF)

m1 <- lmer(devRate~Temp_mean + (1|Site_Code),
           data=modDF)

m2 <- lmer(devRate~RH_mean + (1|Site_Code),
           data=modDF)

m3 <- lmer(devRate~DTR + (1|Site_Code),
           data=modDF)
```

```
m4 <- lmer(devRate~Temp_min + (1|Site_Code),
           data=modDF)

m5 <- lmer(devRate~RH_min + (1|Site_Code),
           data=modDF)

m6 <- lmer(devRate~Temp_max + (1|Site_Code),
           data=modDF)

m7 <- lmer(devRate~RH_max + (1|Site_Code),
           data=modDF)

modelSums <- do.call(rbind, lapply(list(m0, m1,m2,m3, m4, m5, m6, m7), broom::glance))
modelSums #m1 or m4
```

```
##         sigma   logLik       AIC       BIC  deviance df.residual
## 1 0.019768076 153.2228 -300.4457 -294.0163 -316.6063          60
## 2 0.007165630 209.3282 -410.6565 -402.0840 -445.4920          59
## 3 0.009061652 183.4426 -358.8852 -350.3127 -391.4812          59
## 4 0.010435571 183.4782 -358.9563 -350.3838 -391.1167          59
## 5 0.006977285 209.7237 -411.4474 -402.8749 -446.7692          59
## 6 0.010041227 181.0941 -354.1882 -345.6156 -388.6008          59
## 7 0.010883466 180.9413 -353.8826 -345.3101 -385.4466          59
## 8 0.019655919 147.9562 -287.9124 -279.3398 -318.3476          59
```

```
MuMIn::AICc(m0, m1,m2,m3, m4, m5, m6, m7)
```

```
##    df      AICc
## m0  3 -300.0389
## m1  4 -409.9668
## m2  4 -358.1956
## m3  4 -358.2666
## m4  4 -410.7577
## m5  4 -353.4985
## m6  4 -353.1930
## m7  4 -287.2227
```

```
tidy(m1)
```

```
##                       term      estimate    std.error statistic     group
## 1              (Intercept) -6.667066e-02 0.0062175042 -10.72306     fixed
## 2                Temp_mean  5.405793e-03 0.0002666941  20.26964     fixed
## 3 sd_(Intercept).Site_Code  3.571659e-10           NA        NA Site_Code
## 4  sd_Observation.Residual  7.165630e-03           NA        NA  Residual
```

```
#find covariates that aren't correlated
covars <- data.frame(cor(modDF[,2:7]))
rownames(covars[abs(covars$Temp_mean)<0.8,]) #mean RH
```

```
## [1] "RH_mean"
```

```
#model selection
m1 <- lmer(devRate~Temp_mean + (1|Site_Code),
           data=modDF)

m2 <- lmer(devRate~Temp_mean + RH_mean + (1|Site_Code),
           data=modDF)

m3 <- lmer(devRate~Temp_mean * RH_mean + (1|Site_Code),
           data=modDF)

m4 <- lmer(devRate~ RH_mean + (1|Site_Code),
           data=modDF)

modelSums <- do.call(rbind, lapply(list(m1,m2, m3, m4), broom::glance)) #1 seems best
modelSums
```

```
##         sigma   logLik       AIC       BIC  deviance df.residual
## 1 0.007165630 209.3282 -410.6565 -402.0840 -445.4920          59
## 2 0.007155482 201.8216 -393.6431 -382.9275 -445.8598          58
## 3 0.007115474 193.9168 -375.8336 -362.9748 -446.9274          57
## 4 0.009061652 183.4426 -358.8852 -350.3127 -391.4812          59
```

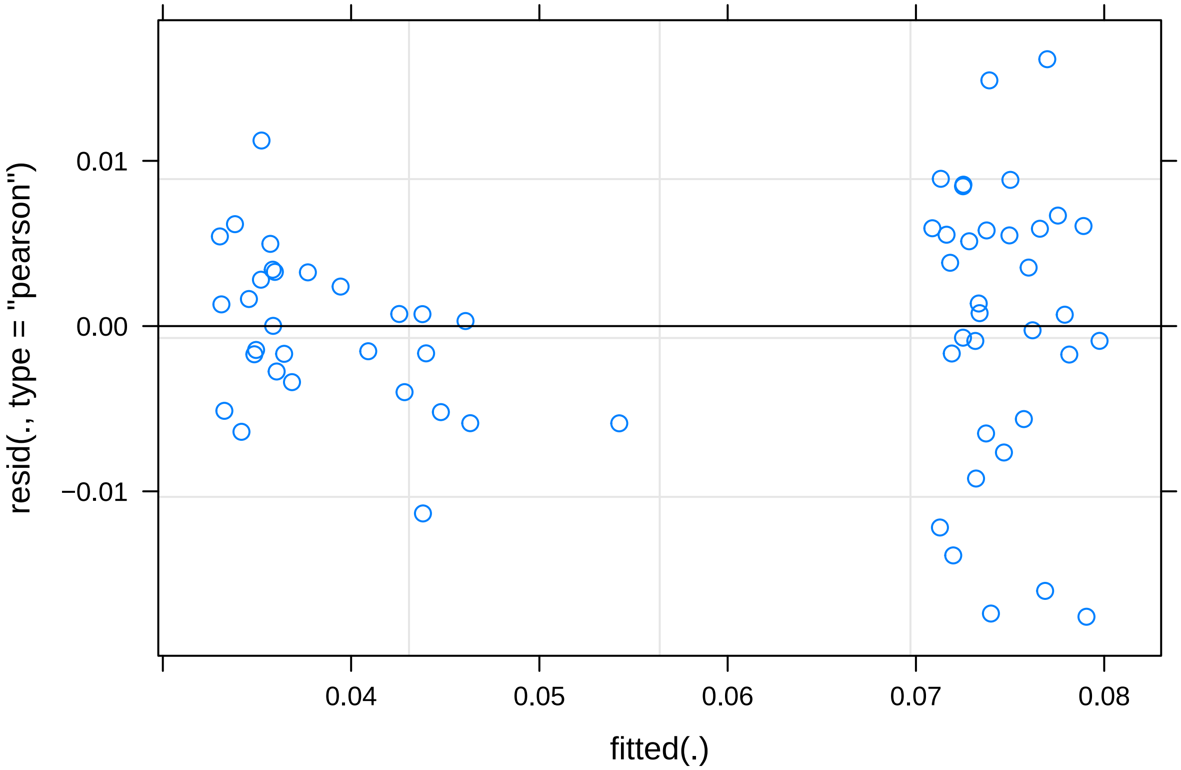
```
AICc(m1,m2, m3, m4)
```

```
##    df      AICc
## m1  4 -409.9668
## m2  5 -392.5905
## m3  6 -374.3336
## m4  4 -358.1956
```

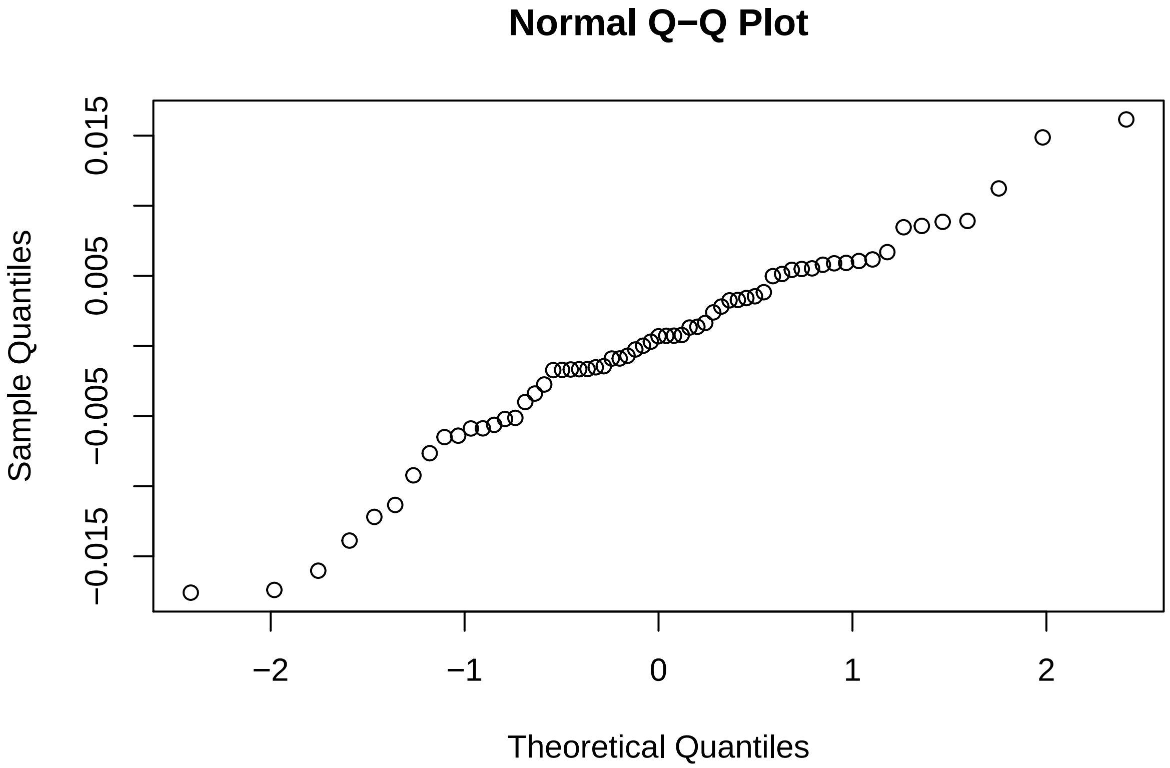
```
tidy(m1)
```

```
##                        term      estimate    std.error statistic     group
## 1              (Intercept) -6.667066e-02 0.0062175042 -10.72306     fixed
## 2                Temp_mean  5.405793e-03 0.0002666941  20.26964     fixed
## 3 sd_(Intercept).Site_Code  3.571659e-10           NA        NA Site_Code
## 4  sd_Observation.Residual  7.165630e-03           NA        NA  Residual
```

```
devClimMod <- lmer(devRate~Temp_mean +(1|Site_Code),
                    data=emergClim)

plot(devClimMod)
```



```
qqnorm(resid(devClimMod))
```



```
summary(devClimMod)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: devRate ~ Temp_mean + (1 | Site_Code)
##    Data: emergClim
##
## REML criterion at convergence: -418.7
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.45479 -0.51575  0.09618  0.73667  2.25385
##
## Random effects:
##  Groups    Name        Variance  Std.Dev.
##  Site_Code (Intercept) 1.276e-19 3.572e-10
##  Residual              5.135e-05 7.166e-03
## Number of obs: 63, groups:  Site_Code, 9
##
## Fixed effects:
##               Estimate Std. Error t value
## (Intercept) -0.0666707  0.0062175  -10.72
## Temp_mean    0.0054058  0.0002667   20.27
##
## Correlation of Fixed Effects:
##           (Intr)
## Temp_mean -0.989
```

```
modResults <- tidy(devClimMod)
#coefficients
modResults$estimate[2] #temp coefficient
```

```
## [1] 0.005405793
```

Indirect Effects

Uninfected Body Size

Load body size data:

```
augWing <- read.csv("../data/emergence/raw/AugustWingLength.csv", stringsAsFactors = F)
octWing <- read.csv("../data/emergence/raw/OctoberWingLength.csv",  stringsAsFactors = F)



#convert to mm & clean
augWing$mm <- augWing$Bars*augWing$Conversion.mm.bars.
octWing$mm <- octWing$Bars*octWing$Conversion.bars.mm.

octWing$site <- as.factor(substr(as.character(octWing$TrayCode), 1, 2))
augWing$site <- as.factor(substr(as.character(augWing$TrayCode), 1, 2))

getClass <- function(monthDf){
  monthDf$class <- NULL
  for (i in 1:nrow(monthDf)){
    if (substr(monthDf$site[i], 1,1)=="R"){
    monthDf$class[i] <- "Rural"
    }
```

```r
    if (substr(monthDf$site[i], 1,1)=="S"){
    monthDf$class[i] <- "Suburban"
    }
    if (substr(monthDf$site[i], 1,1)=="U"){
    monthDf$class[i] <- "Urban"
    }
  }
  monthDf$class <- as.factor(monthDf$class)
  return(monthDf)
}

augWing <- getClass(augWing)
octWing <- getClass(octWing)

octWing$block <- "Fall"
augWing$block <- "Summer"

augWing$Date <- as.Date(as.character(augWing$Date), format="%m/%d/%Y")
octWing$Date <- as.Date(as.character(octWing$Date), format="%m/%d/%Y")

#add day of experiment
augWing$Exp_Day <- as.numeric(augWing$Date-as.Date("2016-08-01", format="%Y-%m-%d"))
octWing$Exp_Day <- as.numeric(octWing$Date-as.Date("2016-09-26", format="%Y-%m-%d"))

#rename traycode column to match
colnames(augWing)[1] <- "Tray_Code"
colnames(octWing)[1] <- "Tray_Code"

#combine
fallWing <- octWing %>%
  dplyr::select(block, class, Site_Code=site, Tray_Code, Exp_Day, mm)

summerWing <- augWing %>%
  dplyr::select(block, class, Site_Code=site, Tray_Code, Exp_Day, mm)

allWing <- rbind(summerWing, fallWing)
allWing$block<- factor(allWing$block, levels=c("Summer", "Fall"))

#drop outlier in S2 (wing size =1.56 mm)
allWing <- allWing %>%
  filter(mm>1.6)
```

Land Class x Season

Model Selection

```r
#model selection by AIC and logLik
m0 <- lmer(mm~1 + (1|Site_Code),
           data=allWing)
m1 <- lmer(mm ~ block + (1|Site_Code),
                   data=allWing)

m2 <- lmer(mm ~ class + (1|Site_Code),
                   data=allWing)

m3 <- lmer(mm ~ class + block + (1|Site_Code),
                   data=allWing)

m4 <- lmer(mm ~ class*block + (1|Site_Code),
                   data=allWing)

modelSums <- do.call(rbind, lapply(list(m0,m1,m2,m3, m4), broom::glance))
modelSums #m0 is best
```

```
##       sigma   logLik       AIC       BIC  deviance df.residual
## 1 0.1782356 80.15274 -154.3055 -143.2243 -165.6758         294
## 2 0.1776557 78.44084 -148.8817 -134.1067 -167.9672         293
## 3 0.1782550 76.79547 -143.5909 -125.1223 -165.8069         292
## 4 0.1776747 75.26388 -138.5278 -116.3654 -168.4225         291
## 5 0.1741417 78.38996 -140.7799 -111.2301 -183.3942         289
```

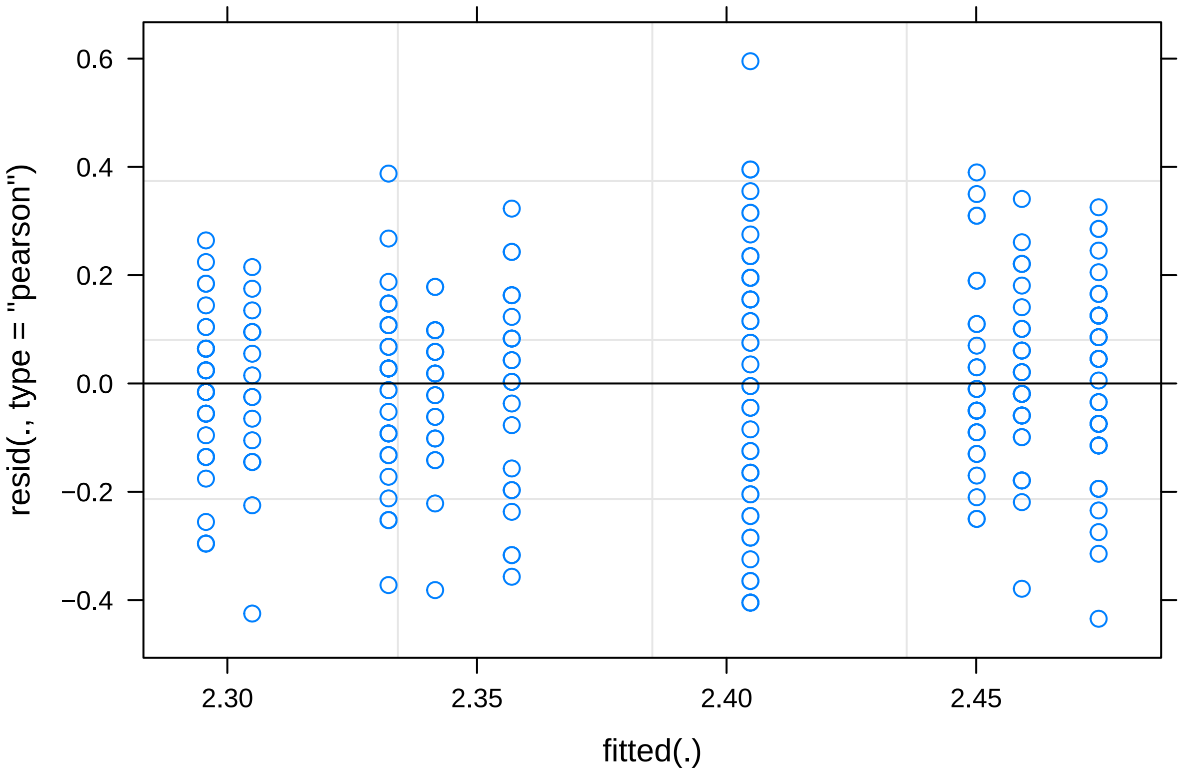
```
AICc(m0, m1,m2,m3, m4) #still m0
```

```
##    df      AICc
## m0  3 -154.2236
## m1  4 -148.7447
## m2  5 -143.3848
## m3  6 -138.2381
## m4  8 -140.2799
```

```
wingMod <- lmer(mm~1 +(1|Site_Code),
                data=allWing)
summary(wingMod)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: mm ~ 1 + (1 | Site_Code)
##    Data: allWing
##
## REML criterion at convergence: -160.3
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.4380 -0.6426  0.1031  0.6165  3.3395
##
## Random effects:
##  Groups    Name        Variance Std.Dev.
##  Site_Code (Intercept) 0.005613 0.07492
##  Residual              0.031768 0.17824
## Number of obs: 297, groups:  Site_Code, 9
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  2.38002    0.02724   87.39
```

```
plot(wingMod)
```



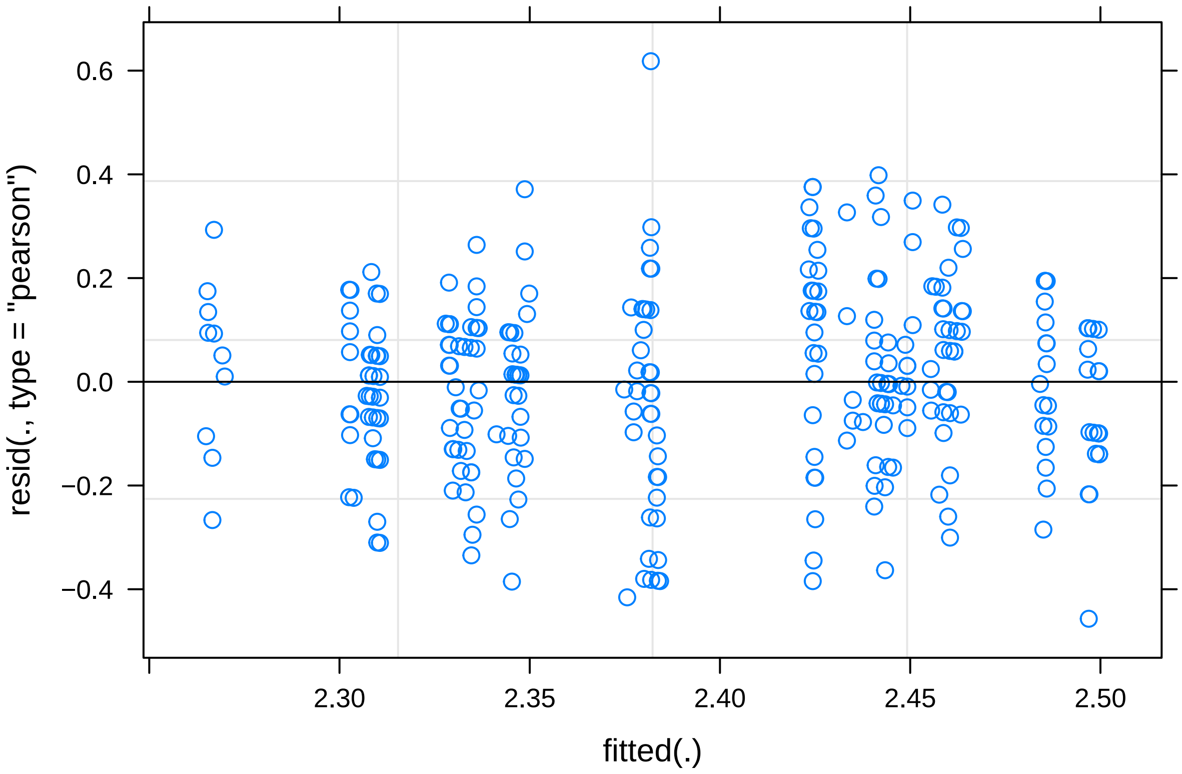
Microclimate

Load Data:

```
climate <- read.csv(file='../data/microclimate/clean/2016TrialsAdultCleaned.csv', stringsAsFactors = F)

climate$Day <- as.Date(climate$Day, format="%Y-%m-%d")
climate$Site_Code <- as.factor(climate$Site_ID)
climate$Tray_Code <- as.factor(climate$Tray_ID)

getClimate <- function(indMosq, climateDF=climate, season){
  #' This is a function to apply over the rows of the octWing and augWing data frames. Must have climat

  #' @param indMosq row of the dataframe for each individual mosquito
  #' @param climateDF the dataframe containing climate data every 10 minutes
  #' @param season, either "summer" or "fall"
  #' @returns formatted data with climate and winglength for the individual mosquito


  #get date range
  startDate <- ifelse(season=="Summer", "2016-08-01", "2016-09-26")
  startDate <- as.Date(startDate, format="%Y-%m-%d")
  endDate <- indMosq$Date

  #subset temperature data
  try(climSubset <- climateDF %>%
    filter(Tray_Code==as.character(indMosq$Tray_Code)) %>%
    filter(Day>startDate & Day<endDate),
    silent=T)

  #now take mean temperature
  tempMean <- climSubset %>%
    summarise(Tmean=mean(Temp, na.rm=T))

  # if (nrow(climSubset)<1000){
  #   climSubset <- climate %>%
  #     filter(Site_ID==indMosq$site) %>%
  #     filter(Day>startDate & Day<endDate)
  # }

  #merge this all together
  #mosqFormat <- cbind(indMosq[,c('block','class','site','Tray_ID','Exp_Day', 'mm')], Tmean=tempMean$Tm

  return(tempMean$Tmean)
}

augWing$Temp <- NA
for(i in 1:nrow(augWing)){
  indMosq <- augWing[i,]
  augWing$Temp[i] <- getClimate(indMosq, season="Summer")
}

octWing$Temp <- NA
for(i in 1:nrow(octWing)){
  indMosq <- octWing[i,]
  octWing$Temp[i] <- getClimate(indMosq, season="Fall")
}

#combine
fallWing <- octWing %>%
  dplyr::select(block, class, Site_Code=site, Tray_Code, Exp_Day, mm, Temp)

summerWing <- augWing %>%
    dplyr::select(block, class, Site_Code=site, Tray_Code, Exp_Day, mm, Temp)

allWing <- rbind(summerWing, fallWing)
allWing$block<- factor(allWing$block, levels=c("Summer", "Fall"))

#drop outlier in S2 (wing size =1.56 mm)
allWing <- allWing %>%
  filter(mm>1.6)
```

Statistics:

```
wingModTemp <- lmer(mm~Temp+(1|Site_Code),
                    data=allWing)
plot(wingModTemp)
```



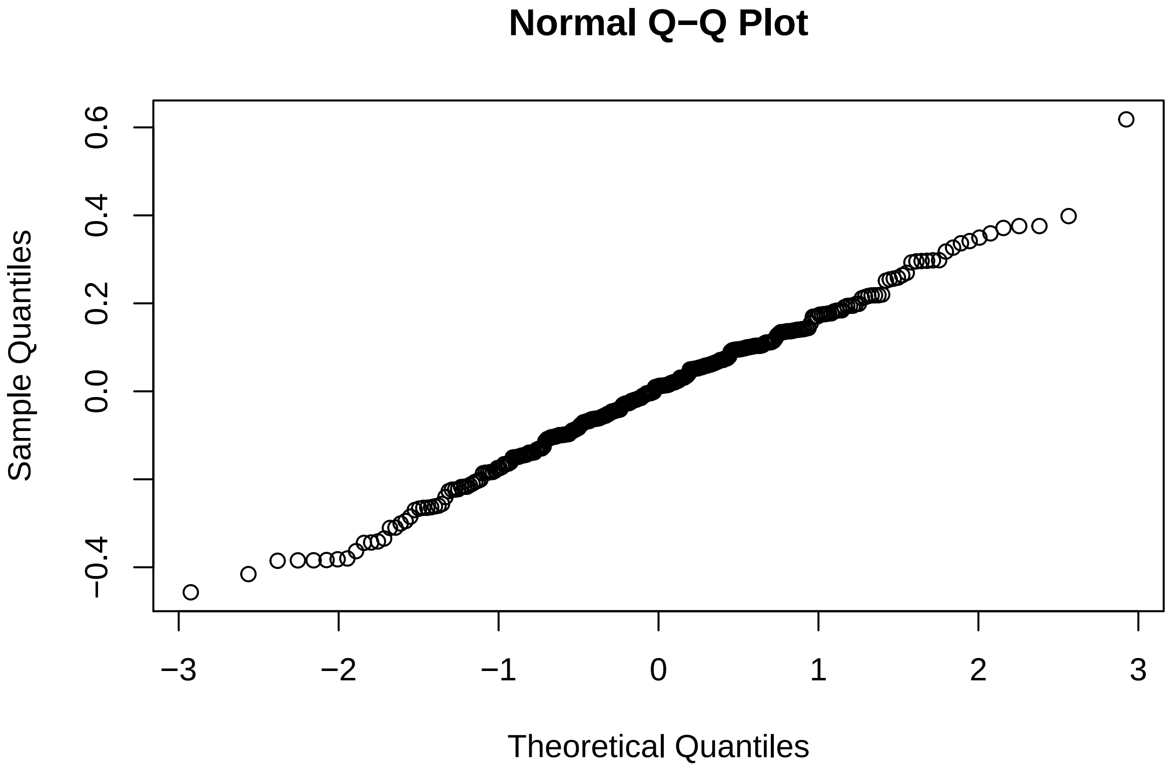
```
summary(wingModTemp)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: mm ~ Temp + (1 | Site_Code)
##    Data: allWing
##
## REML criterion at convergence: -151.5
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.5796 -0.5915  0.0659  0.6242  3.4898
##
## Random effects:
##  Groups    Name        Variance Std.Dev.
##  Site_Code (Intercept) 0.005402 0.0735
##  Residual              0.031376 0.1771
## Number of obs: 290, groups:  Site_Code, 9
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept) 2.253964   0.074292  30.339
## Temp        0.005917   0.003142   1.883
##
## Correlation of Fixed Effects:
##      (Intr)
## Temp -0.932
```

```
confint(wingModTemp) #no effect of temperature
```

```
##                    2.5 %     97.5 %
## .sig01        0.0397126866 0.12624823
## .sigma        0.1631525337 0.19248521
## (Intercept)  2.1094289217 2.40017384
## Temp        -0.0003154784 0.01203926
```

```
qqnorm(resid(wingModTemp))
```



Plot

```
ggplot(data=allWing, aes(x=Temp, y=mm)) +
  geom_point(aes(col=factor(class), shape=factor(block))) +
  geom_smooth(method="lm")+
  theme_fivethirtyeight() +
  theme(panel.background = element_rect(fill = "transparent", colour = NA),
        plot.background = element_rect(fill = "transparent", colour = NA),
        legend.key = element_blank())+
  theme(legend.background = element_rect(fill = "transparent", colour = NA)) +
    theme(axis.title = element_text(), axis.title.x = element_text()) +
  ylab("Uninfected Wing Size")+
  xlab("Mean Temperature (C)") +
  scale_color_manual(values=c(colR, colS, colU), name="Class", labels=c("Rural", "Suburban", "Urban"))+
  #scale_alpha_discrete(range=c(1,0.3), name="Season", labels=c("Fall", "Summer"), guide=guide_legend(r
  scale_shape_discrete(name="Season",labels=c("Fall", "Summer"), guide=guide_legend(reverse=T)) +
  theme(legend.position="right", legend.direction="vertical")+
  guides(shape=F)
```

Growth Rate

Format and merge data

```
#calculate means per Exp_Day and Tray
growthWing <- allWing %>%
  dplyr::group_by(block,class,Site_Code,Tray_Code, Exp_Day) %>%
  dplyr::summarise(meanWing=mean(mm, na.rm=T)) %>%
  dplyr::ungroup()

growthWing$class <- tolower(growthWing$class)

#merge with emergence data
allEmerg$class <- allEmerg$Class
levels(allEmerg$class) <- c("Rural", "Suburban", "Urban")
levels(allEmerg$block) <- c("Summer", "Fall")
growthDF <- merge(allEmerg, growthWing, by=c("block", "class", "Site_Code", "Tray_Code", "Exp_Day"), al

trayMeans <- allWing %>%
  dplyr::group_by(block, Tray_Code) %>%
  dplyr::summarise(meanWing=mean(mm, na.rm=T))

#fill in missing with mean of tray during that block
for (i in 1:nrow(growthDF)){
  if (is.na(growthDF$meanWing[i])){
    temp <- trayMeans$meanWing[trayMeans$block==growthDF$block[i] & trayMeans$Tray_Code==growthDF$Tray_C
    if (length(temp)==0) next
    growthDF$meanWing[i] <- temp
  } else next
}

#some trays had no mosquitoes emerge that weren't infected, so we take the site level mean for them

siteMeans <- allWing %>%
  dplyr::group_by(Site_Code, block) %>%
  dplyr::summarise(meanWing=mean(mm, na.rm=T))

for (i in 1:nrow(growthDF)){
  if (is.na(growthDF$meanWing[i])){
    temp <- siteMeans$meanWing[siteMeans$block==growthDF$block[i] & siteMeans$Site_Code==growthDF$Site_C
    if (length(temp)==0) next
    growthDF$meanWing[i] <- temp
  } else next
}
```

```
growthDF$Fwx <- -121.240 + (78.02 * growthDF$meanWing)
growthDF$AxFwx <- growthDF$Num_Emerge*growthDF$Fwx
growthDF$xAxFwx <- growthDF$Exp_Day*growthDF$AxFwx

#get sum per day
growthDF2 <- growthDF %>%
  dplyr::group_by(block, Tray_Code, class, Site_Code) %>%
  dplyr::summarise(xAxFwx = sum(xAxFwx), AxFwx=sum(AxFwx))

growthDF2 <- growthDF2 %>%
  mutate(r=(log((1/50)*AxFwx))/(14+(xAxFwx/AxFwx))) %>%
  ungroup()
```

Plot

Adjust factor names for plotting:

```r
levels(emergTray$block) <- c("Summer", "Fall")
levels(emergTray$Class) <- c("Rural", "Suburban", "Urban")

levels(survTray$block) <- c("Summer", "Fall")
levels(survTray$Class) <- c("Rural", "Suburban", "Urban")
```

Three panel survival, emergence and growth rate

```r
#png(file="figures/forMS/survEmergeGrowth.png", width = 4, height=7, units="in", res=500, family="sans"

survPlot <- ggplot(data=survTray, aes(x=Class, y=(survival*100)))+
  geom_boxplot(aes(fill=Class), width=0.4)+
  scale_fill_manual(values=c(colR, colS, colU), labels=c("Rural", "Suburban", "Urban"))+
  facet_wrap(~block, ncol=2)+
  ylab("Percent Survival")+
  theme_fivethirtyeight() +
  theme(panel.background = element_rect(fill = "transparent", colour = NA),
        plot.background = element_rect(fill = "transparent", colour = NA),
        legend.key = element_blank(),
        panel.grid.major.x = element_blank(),
        legend.background = element_rect(fill = "transparent", colour = NA),
        axis.title = element_text(),
        axis.title.x = element_text(),
        axis.title.y=element_text(),
        #strip.background = element_blank(),
        strip.text.x = element_text(size=12),
        axis.line=element_line(color=axisColor, size=0.5),
        panel.grid = element_blank(),
        axis.text.y=element_text(size=12),
        axis.text.x=element_text(size=12))+
  theme(axis.title.x = element_blank(),
        axis.text.x = element_blank(),
        axis.title.y= element_text(size=10),
        strip.text = element_text(size=10))+
  theme(legend.position="none")

#Emergence Plot
emergePlot <- ggplot(emergTray, aes(x=Class, y=devRate))+
  geom_boxplot(aes(fill=Class), width=0.4)+
  scale_fill_manual(values=c(colR, colS, colU), labels=c("Rural", "Suburban", "Urban"))+
  facet_wrap(~block, ncol=2)+
  ylab("Development Rate")+
   theme_fivethirtyeight() +
  theme(panel.background = element_rect(fill = "transparent", colour = NA),
        plot.background = element_rect(fill = "transparent", colour = NA),
        legend.key = element_blank(),
        panel.grid.major.x = element_blank(),
        legend.background = element_rect(fill = "transparent", colour = NA),
        axis.title = element_text(),
        axis.title.x = element_text(),
        axis.title.y=element_text(),
        #strip.background = element_blank(),
        strip.text.x = element_text(size=12),
        axis.line=element_line(color=axisColor, size=0.5),
        panel.grid = element_blank(),
        axis.text.y=element_text(size=12),
        axis.text.x=element_text(size=12))+
  theme(legend.position="none") +
  theme(strip.background = element_blank(),
  strip.text.x = element_blank()) +
  theme(axis.title.x = element_blank(),
        axis.text.x = element_blank(),
        axis.title.y= element_text(size=10))

growthPlot <- ggplot(data=growthDF2, aes(x=class, y=r))+
  geom_boxplot(aes(fill=class), width=0.4)+
  scale_fill_manual(values=c(colR, colS, colU), labels=c("Rural", "Suburban", "Urban"))+
  facet_wrap(~block, ncol=2)+
   theme_fivethirtyeight() +
  theme(panel.background = element_rect(fill = "transparent", colour = NA),
        plot.background = element_rect(fill = "transparent", colour = NA),
        legend.key = element_blank(),
        panel.grid.major.x = element_blank(),
        legend.background = element_rect(fill = "transparent", colour = NA),
        axis.title = element_text(),
        axis.title.x = element_text(),
        axis.title.y=element_text(),
        #strip.background = element_blank(),
        strip.text.x = element_text(size=12),
        axis.line=element_line(color=axisColor, size=0.5),
        panel.grid = element_blank(),
        axis.text.y=element_text(size=12),
        axis.text.x=element_text(size=12))+
  theme(legend.position="none") +
  theme(strip.background = element_blank(),
  strip.text.x = element_blank()) +
  ylab("Per Capita Growth\nRate (r')")+
  xlab("Land Class") +
  theme(axis.text.x = element_text(size=10)) +
  theme(axis.title = element_text(size=12),
        axis.title.y= element_text(size=10))

plot_grid(survPlot, emergePlot, growthPlot,
          labels=c("A", "B", "C"),
          nrow=3,
          align='v')

#dev.off()
```

Statistics

Season x Land Class

Model Selection

```
#model selection by AIC and logLik
m1 <- lmer(r ~ block + (1|Site_Code),
                   data=growthDF2)

m2 <- lmer(r ~ class + (1|Site_Code),
                   data=growthDF2)

m3 <- lmer(r ~ block + class + (1|Site_Code),
                   data=growthDF2)

m4 <- lmer(r ~ block*class + (1|Site_Code),
                   data=growthDF2)

modelSums <- do.call(rbind, lapply(list(m1,m2,m3, m4), broom::glance))
modelSums #m1 is best
```

```
##        sigma   logLik       AIC       BIC  deviance df.residual
## 1 0.02317469 150.8876 -293.7752 -284.7812 -319.0258          66
## 2 0.04934191 101.7433 -193.4867 -182.2442 -225.4769          65
## 3 0.02317408 144.3205 -276.6410 -263.1500 -319.4812          64
## 4 0.02304346 138.7162 -261.4324 -243.4445 -322.2346          62
```

```
AICc(m1,m2,m3, m4) #still m1
```

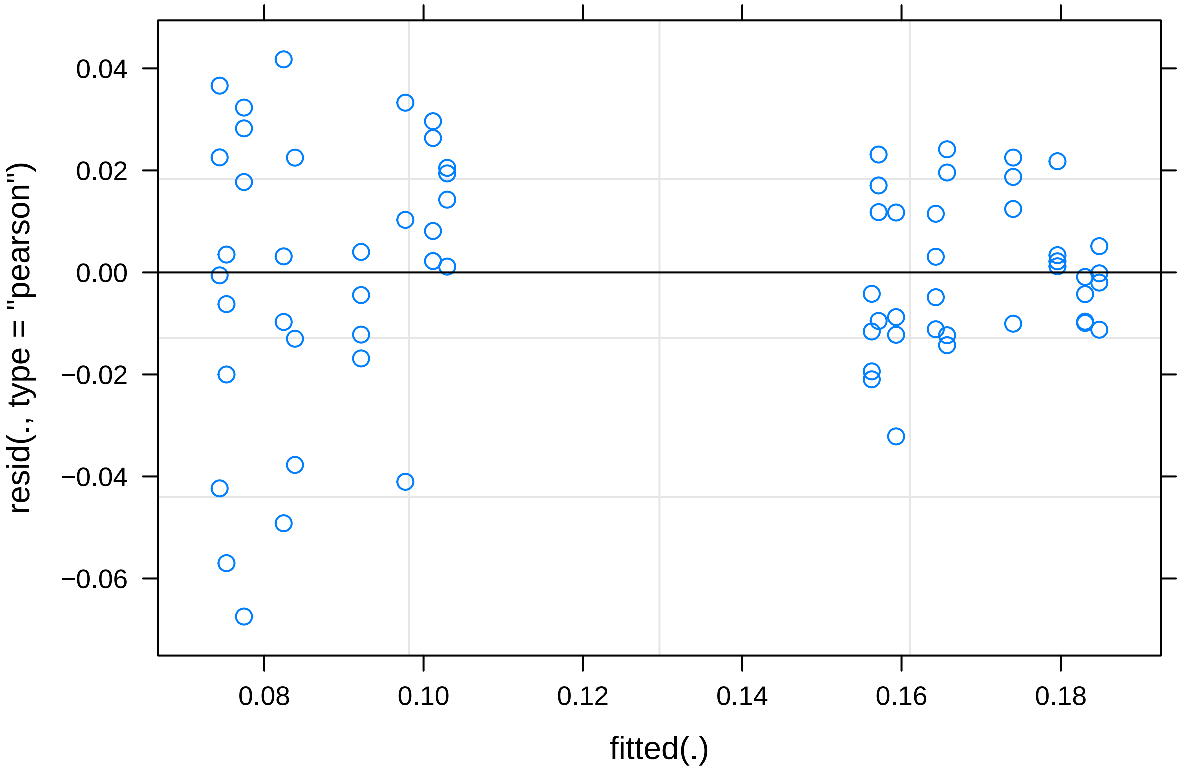
```
##    df      AICc
## m1  4 -293.1598
## m2  5 -192.5492
## m3  6 -275.3076
## m4  8 -259.0718
```

```
growthModSeason <- lmer(r~ block+ (1|Site_Code),
                        data=growthDF2)

summary(growthModSeason)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: r ~ block + (1 | Site_Code)
##    Data: growthDF2
##
## REML criterion at convergence: -301.8
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.91104 -0.48351  0.05068  0.75767  1.80245
##
## Random effects:
##  Groups    Name        Variance  Std.Dev.
##  Site_Code (Intercept) 0.0001764 0.01328
##  Residual              0.0005371 0.02317
## Number of obs: 70, groups:  Site_Code, 9
```

```
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)  0.169354   0.005876   28.82
## blockFall   -0.081859   0.005548  -14.75
##
## Correlation of Fixed Effects:
##           (Intr)
## blockFall -0.458
```

```
plot(growthModSeason)
```



```
tidy(growthModSeason)
```

```
##                        term     estimate   std.error statistic     group
## 1               (Intercept)  0.16935442 0.005875706  28.82282     fixed
## 2                 blockFall -0.08185939 0.005548448 -14.75356     fixed
## 3 sd_(Intercept).Site_Code  0.01328340          NA        NA Site_Code
## 4  sd_Observation.Residual  0.02317469          NA        NA  Residual
```

```
car::Anova(growthModSeason) #Wald test
```

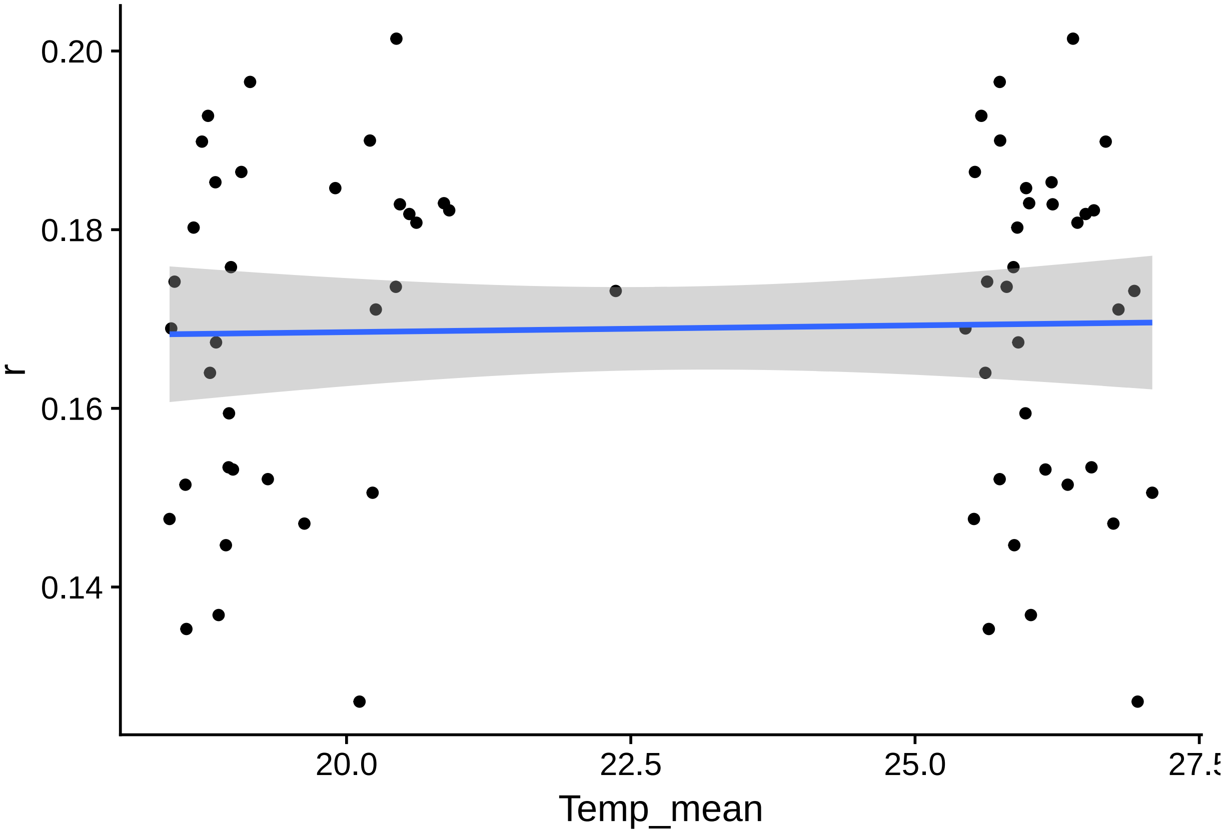
```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: r
##        Chisq Df Pr(>Chisq)
## block 217.67  1  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
confint(growthModSeason) #profiled confidence interval
```

```
##                    2.5 %      97.5 %
## .sig01       0.005098632  0.02410232
## .sigma       0.019439256  0.02774893
## (Intercept)  0.157555871  0.18115298
## blockFall   -0.092829521 -0.07091747
```

Microclimate

```
levels(allClim$block) <- factor(as.character(allClim$block), levels=c("Summer", "Fall"))
#get temperature data
growthTemp <- merge(growthDF2, allClim, by.x=c("block","Tray_Code"), by.y=c("block", "Tray_ID"))
ggplot(data=growthTemp, aes(x=Temp_mean, y=r))+
  geom_point()+
  geom_smooth(method="lm")
```



```
growthTempMod <- lm(r~Temp_mean,
                    data=growthTemp)
summary(growthTempMod)
```

```
##
## Call:
## lm(formula = r ~ Temp_mean, data = growthTemp)
##
## Residuals:
```

```
##       Min        1Q    Median        3Q       Max 
## -0.042414 -0.016481  0.004224  0.014253  0.032784 
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 0.1655056  0.0158639   10.43 1.34e-15 ***
## Temp_mean   0.0001513  0.0006878    0.22    0.827    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.01919 on 66 degrees of freedom
## Multiple R-squared:  0.0007328,  Adjusted R-squared:  -0.01441 
## F-statistic: 0.0484 on 1 and 66 DF,  p-value: 0.8266
```

```
anova(growthTempMod)
```

```
## Analysis of Variance Table
## 
## Response: r
##           Df    Sum Sq   Mean Sq F value Pr(>F)
## Temp_mean  1 0.0000178 1.782e-05  0.0484 0.8266
## Residuals 66 0.0243013 3.682e-04
```

Vectorial Capacity

Arrange temperature data by hour for rate summation

```
climS <- climate[climate$Day >="2016-08-01" & climate$Day <="2016-09-03",]
climS$Block <- "summer"

climF <- climate[climate$Day >="2016-09-26" & climate$Day <="2016-11-08",]
climF$Block <- "fall"

#bind back together
climAll <- rbind(climS, climF)
climAll$Date <- as.POSIXct(climAll$Date)
climAll$Hour <- lubridate::hour(climAll$Date)

parameters <- dplyr::select(climAll, Block, Class, Site_ID, Tray_ID, Temp, Day, Hour)
```

Calculated set at 27C constant (x=27):

Briere: y ~ a * x * (x - t0) * (tmax - x)^(1/2) Quad: y ~ a * (x-t0) * (x-Tmax)

```
parameters <- unique(dplyr::select(climAll, Block, Class, Site_ID))
#a -- bite rate
parameters$a <- ((1.93/10000)*27*(27-10.25)*((38.32-27)^0.5))
#adjust negatives
parameters$a[parameters$a<0] <- 0
parameters$a[is.na(parameters$a)] <- 0

# PDR - parasite development rate
parameters$PDR <- ((1.09/10000)*27*(27-10.39)*((43.05-27)^0.5))
#adjust negatives
parameters$PDR[parameters$PDR<0] <- 0
```

```r
parameters$PDR[is.na(parameters$PDR)] <- 0

# lf - mosquito lifespan
parameters$lf <- -1.43*(27- 13.41)*(27-31.51)
#adjust for zeros
parameters$lf[parameters$lf<0] <- 0
parameters$lf[is.na(parameters$lf)] <- 0

# we will then calculate the mean and se for these parameters per site and hour, then sum them up for a
library(dplyr)

paramRate <- parameters

paramRate <- dplyr::rename(paramRate, Site_Code=Site_ID)
paramRate <- dplyr::rename(paramRate, block=Block)

## u - daily probability of mosquito mortality
paramRate$mu <- 1/paramRate$lf

paramRate$block <- tolower(paramRate$block)
paramRate$Class <- tolower(paramRate$Class)

paramRateOld <- paramRate
```

Combine with field measured fecundity, survival, development rate and vector competence:

```r
#EFD: from wing length
EFD  <- allWing %>%
  group_by(block, Class=class, Site_Code) %>%
  dplyr::summarise(wingL=mean(mm, na.rm=T)) %>%
  ungroup() %>%
  mutate(fecundity=-121.240 + (78.02*wingL))
EFD$block <- tolower(EFD$block)
EFD$Class <- tolower(EFD$Class)

#pEA: larval survival
pEA <- survSumm %>%
  group_by(block, Class, Site_Code) %>%
  dplyr::summarise(pEA=mean((percSurv/50), na.rm=T))
pEA$block <- tolower(pEA$block)
pEA$Class <- tolower(pEA$Class)

#MDR: emergence rate (day^-1)
MDR <- emergTray %>%
  group_by(block, Class, Site_Code) %>%
  dplyr::summarise(MDR=mean(devRate, na.rm=T))
MDR$block <- tolower(MDR$block)
MDR$Class <- tolower(MDR$Class)

#bc
bc <- seasonInfSite %>%
  dplyr::select(block, Class=class, Site_Code=site, bc=Saliva_mean)
bc$block <- tolower(bc$block)
bc$Class <- tolower(bc$Class)
```

Merge traits together

```
paramRate <- full_join(paramRate, EFD, by=c("block", "Class", "Site_Code"))
paramRate <- full_join(paramRate, pEA, by=c("block", "Class", "Site_Code"))
paramRate <- full_join(paramRate, MDR, by=c("block", "Class", "Site_Code"))
paramRate <- full_join(paramRate, bc, by=c("block", "Class", "Site_Code"))
```

Calculate EFD

```
paramRate$EFD <- paramRate$fecundity*paramRate$a
```

Calcuate VC w/o carry-over effects

```
paramNoCOE <- dplyr::select(climAll, Block, Class, Site_ID, Tray_ID, Temp, Day, Hour)

#based on model
paramNoCOE$fecundity2 <- ((4.88/100)*27*(27-8.02)*((35.65-27)^0.5))/24
paramNoCOE$fecundity2[paramNoCOE$fecundity2<0] <- 0
paramNoCOE$fecundity2[is.na(paramNoCOE$fecundity2)] <- 0

paramNoCOE$bc2 <- (((7.35/10000)*27*(27-15.84)*((36.40-27)^0.5)) * ((4.39/10000)*27*(27-3.62)*((36.82-2
paramNoCOE$bc2[paramNoCOE$bc2<0] <- 0
paramNoCOE$bc2[is.na(paramNoCOE$bc2)] <- 0


#take mean of each hour and sum up
library(dplyr)
paramRateNoCOE <- paramNoCOE %>%
  group_by(Block, Class, Site_ID, Hour) %>%
  dplyr::select(-Temp, -Day, -Tray_ID) %>%
  summarise_all(funs(mean(.,na.rm=T))) %>%
  ungroup() %>%
  group_by(Block, Class, Site_ID) %>%
  summarise_all(funs(sum)) %>%
  ungroup()

paramRateNoCOE <- dplyr::rename(paramRateNoCOE, Site_Code=Site_ID)
paramRateNoCOE <- dplyr::rename(paramRateNoCOE, block=Block)

paramRateNoCOE$Class <- tolower(paramRateNoCOE$Class)
```

Merge w/ and w/o carry-over effects together

```
paramAll <- full_join(paramRate, paramRateNoCOE, by=c("block", "Class", "Site_Code"))

#calculate EFD
paramAll$EFD2 <- paramAll$fecundity2*paramAll$a
```

Calculate VC from traits (VCnoCOE is w/o carry-over)

```
paramAll <- mutate(paramAll,
                    VC=((a^2)*bc*(exp(-mu/PDR))*EFD*pEA*(MDR^2))/((mu^2)))

paramAll <- mutate(paramAll,
                   VCnoCOE=(((a^2)*bc2*(exp(-mu/PDR))*EFD2*pEA*(MDR^2))/((mu^2))))
```

Plotting

Add Temperature Back in For Comparison Plots

```
siteTemps <- dplyr::select(seasonInfSite, block, site, class, Temp_mean_mean)
levels(paramAll$Class) <- c("Rural", "Suburban", "Urban")
VCplot <- merge(paramAll, siteTemps, by.x=c("block", "Site_Code"), by.y=c("block", "site"))
VCplot <- VCplot %>%
  dplyr::select(block, Class, Site_Code, VC, VCnoCOE, Temp_mean_mean) %>%
  mutate(VCdiff=(VC-VCnoCOE)/VCnoCOE*100)
VCplot$block <- factor(VCplot$block, levels=c("summer", "fall"))
```

Comparison Line Plot

```
#pdf(file="figures/forMS/VCxTemp.pdf", width = 4, height=4, family="sans")

VCxTempPlot <- ggplot(data=VCplot, aes(x=Temp_mean_mean))+
  geom_smooth(aes(y=VC, color="WithCOE"), method="lm", show.legend=F)+
  geom_smooth(aes(y=VCnoCOE, color="WithoutCOE"),  method="lm", show.legend=F)+
  geom_smooth(aes(y=VC, color="WithCOE"), method="lm", fill=NA)+
  geom_smooth(aes(y=VCnoCOE, color="WithoutCOE"),  method="lm", fill=NA)+
  geom_point(aes(y=VC, color="WithCOE")) +
  geom_point(aes(y=VCnoCOE, color="WithoutCOE"))+
  coord_cartesian(ylim=c(0,40))+
  theme_fivethirtyeight()+
  theme(panel.background = element_rect(fill = "transparent", colour = NA),
        plot.background = element_rect(fill = "transparent", colour = NA),
        legend.key = element_blank(),
        legend.background = element_rect(fill = "transparent", colour = NA),
        legend.position="bottom",
        legend.direction="horizontal",
        axis.title = element_text(),
        legend.text=element_text(size=8),
        legend.title = element_text(size=10),
        axis.line=element_line(color="gray40", size=0.5),
        panel.grid = element_blank())+
  guides(colour=guide_legend(title.position="top", title.hjust=0.5))+
  xlab("Temperature (C)")+
  ylab("Vectorial Capacity")+
  scale_colour_manual(name="Calcuation Type",
                      #values=c(WithCOE="#af8dc3", WithoutCOE="#7fbf7b"),
                      values=c(WithCOE="black", WithoutCOE="gray55"),
                      labels=c("With COEs", "Without COEs"))
VCxTempPlot
#dev.off()
```

```
#get mean and se summary
VCplotsumm <- VCplot %>%
  dplyr::select(-Site_Code) %>%
  gather(calc, value, VC, VCnoCOE, VCdiff) %>%
  group_by(block, Class, calc) %>%
  dplyr::summarise_all(funs(mean=mean(., na.rm=T), se=sd(.)/sqrt(n()))) %>%
  ungroup()
#add in facet labels
VCplotsumm$labels <- case_when(
  VCplotsumm$calc=="VC" ~ "With COEs" ,
  VCplotsumm$calc=="VCnoCOE" ~ "No COEs",
  VCplotsumm$calc=="VCdiff" ~ "Difference due to COEs"
)
VCplotsumm$labels <- factor(VCplotsumm$labels, levels=c("No COEs", "With COEs", "Difference due to COEs
#order factor
VCplotsumm$calc <- factor(VCplotsumm$calc, levels=c("VCnoCOE", "VC", "VCdiff"))
```

Panel Plots with Bar Graphs

```
noCOEplot <- ggplot(data=VCplotsumm[VCplotsumm$labels=="No COEs",], aes(x=factor(Class), group=block)) +
  geom_bar(stat='identity',
           aes(y=value_mean, fill=Class, alpha=block, color=Class),
           position=position_dodge(0.9),
           width=0.7) +
  geom_errorbar(aes(ymin=value_mean-value_se, ymax=value_mean+value_se),
                position=position_dodge(0.9),
                width=0.4,
                color=errorColor)+
  ylim(0,50)+
  #facet_wrap(~labels, nrow=3) +
  #theme_fivethirtyeight() +
  ylab("Vectorial Capacity")+
  xlab("Land Class")+
  #theme_fivethirtyeight() +
  scale_x_discrete(labels=c("Rural", "Suburban", "Urban"))+
  scale_fill_manual(values=c(colR, colS, colU),
                    labels=c("Rural", "Suburban", "Urban")) +
  scale_color_manual(values=c(colR, colS, colU))+
  scale_alpha_discrete(range=c(1,0),
                       name="Season",
                       guide=guide_legend(),
                       labels=c("Summer", "Fall"))+
  #geom_hline(aes(yintercept=0))+
  guides(fill=F, alpha=F, color=F)+
  theme(panel.background = element_rect(fill = "transparent", colour = NA),
        plot.background = element_rect(fill = "transparent", colour = NA),
        legend.key = element_blank(),
        legend.background = element_rect(fill = "transparent", colour = NA),
        legend.position="right",
        axis.title.y=element_text(size=12),
        axis.title.x = element_blank(),
        axis.text.x = element_text(size=10),
        legend.text=element_text(size=6),
        legend.title = element_text(size=8),
        legend.key.size = unit(0.1, "in"),
        legend.direction = "vertical",
        legend.box = "vertical",
        panel.grid.major.x = element_blank(),
        panel.grid.major=element_blank(),
        strip.background = element_blank(),
        axis.ticks=element_blank(),
        axis.line=element_line(color=axisColor)) +
 theme(strip.text.x = element_text()) #this has to be seperate for some reason?
```

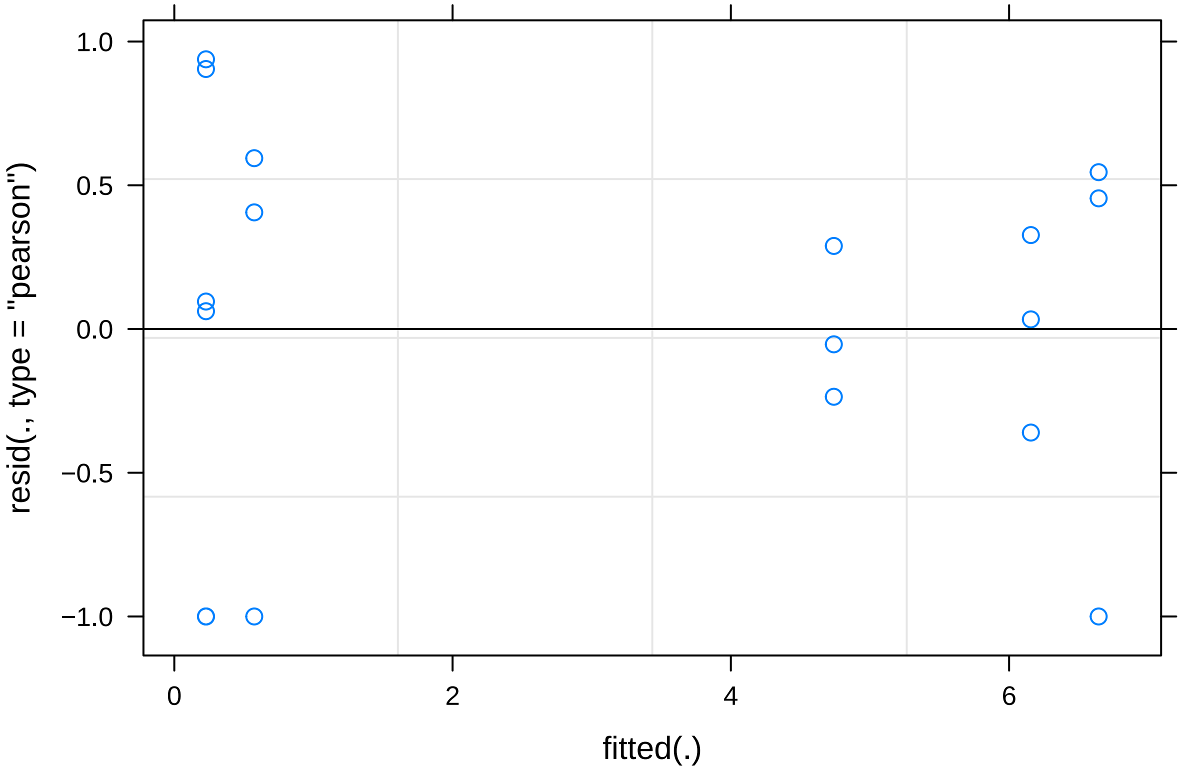
```
noCOEplot

withCOEplot <- ggplot(data=VCplotsumm[VCplotsumm$labels=="With COEs",], aes(x=factor(Class), group=bloc
  geom_bar(stat='identity',
           aes(y=value_mean, fill=Class, alpha=block, color=Class),
           position=position_dodge(0.9),
           width=0.7) +
  geom_errorbar(aes(ymin=value_mean-value_se, ymax=value_mean+value_se),
                position=position_dodge(0.9),
                width=0.4,
                color=errorColor)+
  ylim(0,50)+
  #facet_wrap(~labels, nrow=3) +
  ylab("Vectorial Capacity")+
  xlab("Land Class")+
  #theme_fivethirtyeight() +
  scale_x_discrete(labels=c("Rural", "Suburban", "Urban"))+
  scale_fill_manual(values=c(colR, colS, colU),
                    labels=c("Rural", "Suburban", "Urban")) +
  scale_color_manual(values=c(colR, colS, colU))+
  scale_alpha_discrete(range=c(1,0),
                       name="Season",
                       guide=guide_legend(),
                       labels=c("Summer", "Fall"))+
  #geom_hline(aes(yintercept=0))+
  guides(color=F, fill=F, alpha=F)+
  theme(panel.background = element_rect(fill = "transparent", colour = NA),
        plot.background = element_rect(fill = "transparent", colour = NA),
        legend.key = element_blank(),
        legend.background = element_rect(fill = "transparent", colour = NA),
        legend.position="right",
        axis.title.y=element_text(size=12),
        axis.title.x = element_blank(),
        axis.text.x = element_text(size=10),
        legend.text=element_text(size=6),
        legend.title = element_text(size=8),
        legend.key.size = unit(0.1, "in"),
        legend.direction = "vertical",
        legend.box = "vertical",
        panel.grid.major.x = element_blank(),
        panel.grid.major=element_blank(),
        strip.background = element_blank(),
        axis.line = element_line(color=axisColor),
        axis.ticks=element_blank()) +
 theme(strip.text.x = element_text()) #this has to be seperate for some reason?

withCOEplot

diffPlot <- ggplot(data=VCplotsumm[VCplotsumm$labels=="Difference due to COEs",], aes(x=factor(Class),
  geom_bar(stat='identity',
           aes(y=value_mean, fill=Class, alpha=block, color=Class),
           position=position_dodge(0.9),
           width=0.7) +
  geom_errorbar(aes(ymin=value_mean-value_se, ymax=value_mean+value_se),
                position=position_dodge(0.9),
                width=0.4,
                color=errorColor)+
  #theme_fivethirtyeight() +
  ylab("Change in VC") +
  xlab("Land Class") +
  scale_x_discrete(labels=c("Rural", "Suburban", "Urban"))+
  scale_fill_manual(values=c(colR, colS, colU),
                    labels=c("Rural", "Suburban", "Urban")) +
  scale_color_manual(values=c(colR, colS, colU))+
  scale_alpha_discrete(range=c(1,0),
                       name="Season",
                       guide=guide_legend(),
                       labels=c("Summer", "Fall"))+
  geom_hline(aes(yintercept=0))+
  guides(color=F, alpha=guide_legend(override.aes=list(color=axisColor)))+
  theme(panel.background = element_rect(fill = "transparent", colour = NA),
        plot.background = element_rect(fill = "transparent", colour = NA),
        legend.key = element_blank(),
        legend.background = element_rect(fill = "transparent", colour = NA),
        legend.position="bottom",
        axis.title.y = element_text(size=12),
        axis.title.x = element_text(size=12),
        axis.text.x = element_text(size=10),
        legend.text=element_text(size=8),
        legend.title = element_text(size=12),
        legend.key.size = unit(0.1, "in"),
        legend.direction = "horizontal",
        legend.box = "vertical",
        panel.grid.major.x = element_blank(),
        panel.grid.major=element_blank(),
        strip.background = element_blank(),
        axis.ticks=element_blank(),
        axis.line = element_line(color=axisColor)) +
 theme(strip.text.x = element_text()) #this has to be seperate for some reason?
diffPlot
```

Use cowplot to combine the three bar graphs and VC over temp chart.

```
#pdf(file="figures/forMS/VCPlot.pdf", width = 10, height=7, family="sans")

ggdraw()+
  draw_plot(VCxTempPlot, x=0, y=0, width=0.7, height=1)+
  draw_plot(noCOEplot, x=0.7, y=0.72, width=0.3, height=0.28)+
  draw_plot(withCOEplot, x=0.7, y=0.44, width=0.3, height=0.28)+
  draw_plot(diffPlot, x=0.685, y=0, width=0.3, height=0.44)+
  draw_plot_label(label=c("A", "B", "C", "D"),
                  x=c(0,0.67,0.67,0.67),
                  y=c(1,1,0.72,0.44))
#dev.off()
```

Statistics

```
VCplot$VCforStats <- VCplot$VC+0.000000001
VCplot$VCforStatsnoCOE <- VCplot$VCnoCOE+0.000000001
VCclass <- glmer(VCforStats~Class*block+ (1|Site_Code),
                 data=VCplot,
                family=Gamma())

plot(VCclass)
```



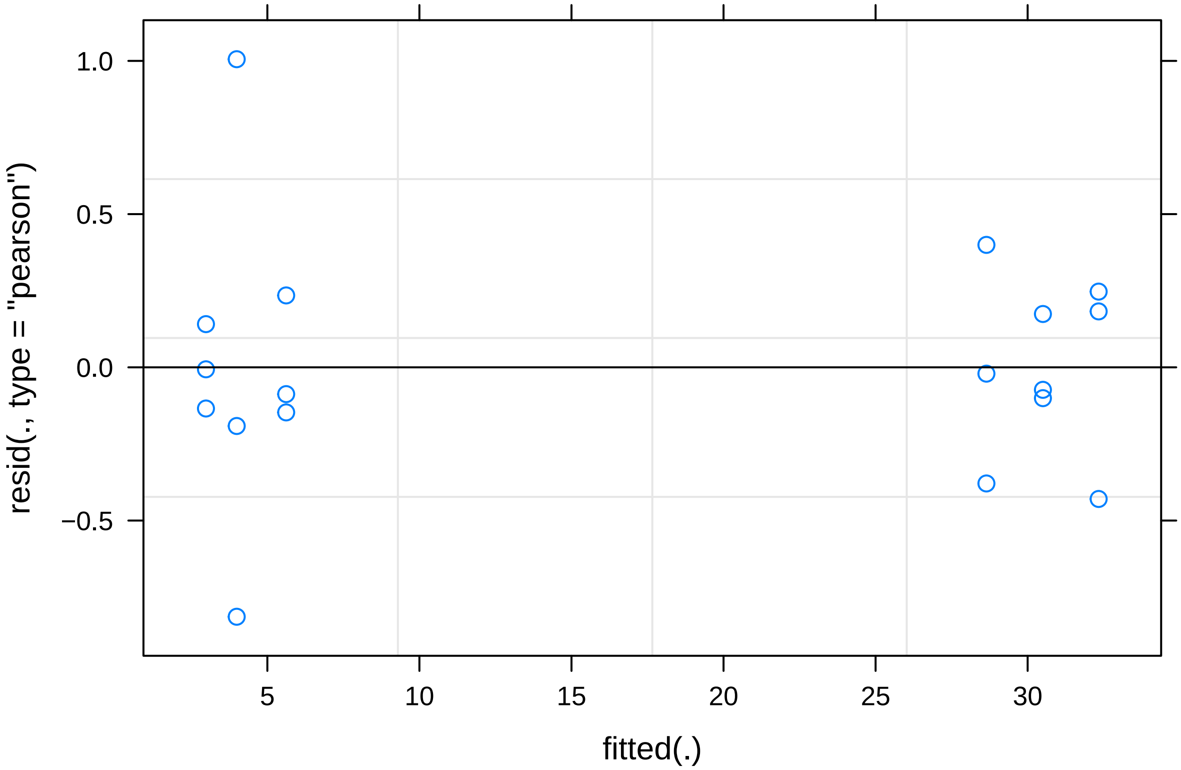
```
summary(VCclass)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: Gamma  ( inverse )
## Formula: VCforStats ~ Class * block + (1 | Site_Code)
##    Data: VCplot
## 
##      AIC      BIC   logLik deviance df.resid 
##    -21.3    -14.2     18.6    -37.3       10 
## 
## Scaled residuals: 
##     Min      1Q  Median      3Q     Max 
## -1.5908 -0.5236  0.1252  0.7036  1.4925 
## 
## Random effects:
##  Groups    Name        Variance Std.Dev.
##  Site_Code (Intercept) 0.0000   0.0000  
##  Residual              0.3951   0.6286  
## Number of obs: 18, groups:  Site_Code, 9
##
## Fixed effects:
##                         Estimate Std. Error t value Pr(>|z|)
## (Intercept)              0.21097    0.28460   0.741    0.459
## Classsuburban           -0.04853    0.35919  -0.135    0.893
## Classurban              -0.06044    0.34962  -0.173    0.863
## blockfall                4.18018    5.93087   0.705    0.481
## Classsuburban:blockfall  0.05330    8.39037   0.006    0.995
## Classurban:blockfall    -2.59022    6.38198  -0.406    0.685
##
## Correlation of Fixed Effects:
##             (Intr) Clsssb Clssrb blckfl Clsss:
## Classsubrbn -0.792
## Classurban  -0.814  0.645
## blockfall   -0.048  0.038  0.039
## Clsssbrbn:b  0.034 -0.043 -0.028 -0.707
## Clssrbn:blc  0.045 -0.035 -0.055 -0.929  0.657
```

```
car::Anova(VCclass)
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: VCforStats
##              Chisq Df Pr(>Chisq)
## Class       0.0400  2     0.9802
## block       1.1650  1     0.2804
## Class:block 0.2958  2     0.8625
```

```
VCclassnoCOE <- glmer(VCforStatsnoCOE~Class*block+ (1|Site_Code),
                 data=VCplot,
                family=Gamma())

plot(VCclassnoCOE)
```



```
summary(VCclassnoCOE)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: Gamma  ( inverse )
## Formula: VCforStatsnoCOE ~ Class * block + (1 | Site_Code)
##    Data: VCplot
##
##      AIC      BIC   logLik deviance df.resid
##    122.2    129.3    -53.1    106.2       10
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.2014 -0.3893 -0.1272  0.4879  2.7192
##
## Random effects:
##  Groups    Name        Variance Std.Dev.
##  Site_Code (Intercept) 0.0000   0.0000
##  Residual              0.1367   0.3697
## Number of obs: 18, groups:  Site_Code, 9
##
## Fixed effects:
##                            Estimate Std. Error t value Pr(>|z|)
## (Intercept)                0.032784   0.007897   4.152  3.3e-05 ***
## Classsuburban              0.002126   0.011536   0.184 0.853751
## Classurban                -0.001858   0.010856  -0.171 0.864088
## blockfall                  0.302685   0.081193   3.728 0.000193 ***
## Classsuburban:blockfall -0.087132   0.101503  -0.858 0.390666
## Classurban:blockfall      -0.155630   0.092118  -1.689 0.091132 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) Clsssb Clssrb blckfl Clsss:
## Classsubrbn -0.685
## Classurban  -0.727  0.498
## blockfall   -0.097  0.067  0.071
## Clsssbrbn:b  0.078 -0.114 -0.057 -0.800
## Clssrbn:blc  0.086 -0.059 -0.118 -0.881  0.705
```

```
car::Anova(VCclassnoCOE)
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: VCforStatsnoCOE
##               Chisq Df Pr(>Chisq)
## Class        0.2985  2     0.8614
## block       34.7659  1  3.718e-09 ***
## Class:block  3.0744  2     0.2150
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#make long for comparison of with and without carry-over
VClong <- VCplot %>%
  gather(COE, value, VC, VCnoCOE)

VCtemp <- lm(value~Temp_mean_mean*COE, data=VClong)
#plot(VCtemp)
summary(VCtemp)
```

```
##
## Call:
## lm(formula = value ~ Temp_mean_mean * COE, data = VClong)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.3372  -1.3931   0.4208   1.1874  10.8765
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)               -15.2797     8.1697  -1.870   0.0706 .
## Temp_mean_mean              0.8017     0.3527   2.273   0.0299 *
## COEVCnoCOE                -57.0284    11.5537  -4.936 2.39e-05 ***
## Temp_mean_mean:COEVCnoCOE   3.1099     0.4988   6.234 5.52e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.984 on 32 degrees of freedom
## Multiple R-squared:  0.8631, Adjusted R-squared:  0.8502
## F-statistic: 67.24 on 3 and 32 DF,  p-value: 6.578e-14
```

```
## Vectorial Capacity (w carry over) across Temp
VCtempCOE <- lm(VCforStats~Temp_mean_mean, data=VCplot)
#plot(VCtempCOE)
summary(VCtempCOE)
```

```
##
## Call:
## lm(formula = VCforStats ~ Temp_mean_mean, data = VCplot)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.2585 -1.3362  0.2381  0.8065  4.4154
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -15.2797     3.9592  -3.859 0.001388 **
## Temp_mean_mean   0.8017     0.1709   4.690 0.000246 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.415 on 16 degrees of freedom
## Multiple R-squared:  0.5789, Adjusted R-squared:  0.5526
## F-statistic: 21.99 on 1 and 16 DF,  p-value: 0.000246
```