

Residence time and microcystis

Rosie

2023-02-17

Residence time data

```
#Residence time from DSM2 models
```

```
library(tidyverse)
```

```
## — Attaching packages — tidyverse 1.3.2 —
## ✓ ggplot2 3.4.1      ✓ purrr  1.0.1
## ✓ tibble  3.1.8      ✓ dplyr  1.1.0
## ✓ tidyr   1.3.0      ✓ stringr 1.5.0
## ✓ readr   2.1.4      ✓ forcats 0.5.2
## — Conflicts — tidyverse_conflicts() —
## ✗ dplyr::filter() masks stats::filter()
## ✗ dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
##
## The following objects are masked from 'package:base':
##
##   date, intersect, setdiff, union
```

```
library(sf)
```

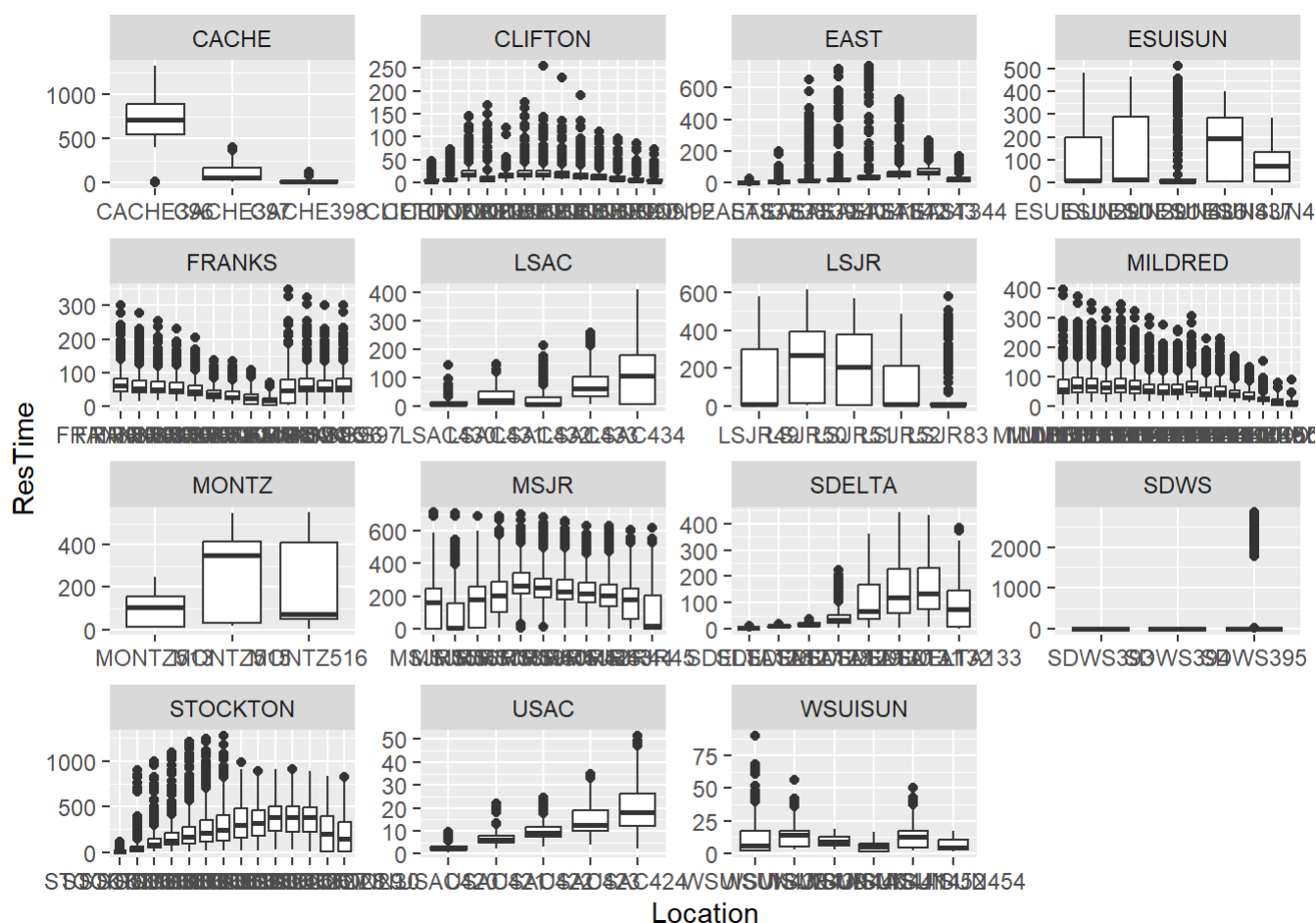
```
## Linking to GEOS 3.9.3, GDAL 3.5.2, PROJ 8.2.1; sf_use_s2() is TRUE
```

```
#read in the residence time data
```

```
res1 = read_csv("data/EX_2020_locRT.csv") %>%
  pivot_longer(cols = -SimPeriod, names_to = "Location", values_to = "ResTime") %>%
  mutate(Date = as.Date(SimPeriod, format = "%d%b%Y"), Region = str_extract(Location, "[A-Z]+"
  ))
```

```
## Rows: 558 Columns: 120
## — Column specification —————
## Delimiter: ","
## chr (1): SimPeriod
## dbl (119): SDELTA126, SDELTA127, SDELTA128, SDELTA129, SDELTA130, SDELTA131,...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
ggplot(res1, aes(x = Location, y = ResTime)) + geom_boxplot()+facet_wrap(~Region, scales = "free")
```



OK, so most regions all have fairly similar residence times per reach. But let's look at it by WYT just for funsies

```
Yrs = read_csv("data/yearassignments.csv") %>%
  select(Year, Yr_type, Index)
```

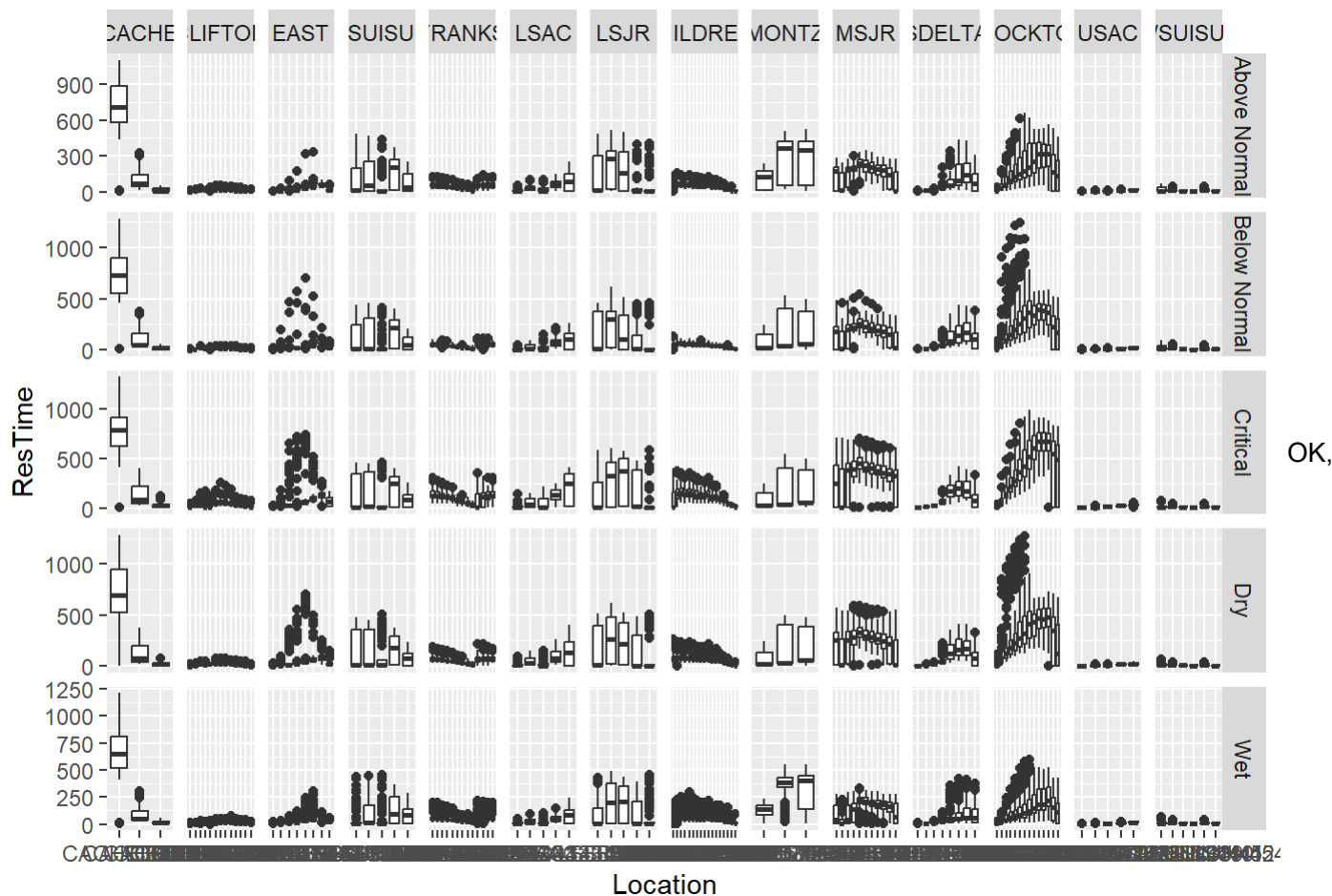
```
## Rows: 117 Columns: 8
## — Column specification —————
## Delimiter: ","
## chr (5): Yr_type, Drought, DroughtYear, ShortTerm, Whitepaper
## dbl (3): Year, Index, SprNDOI
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

*#In other circumstances I'd merge these by water years, but the october of a year
#after a dry summer is probalby more like September of the same year than it is like February of
the
#next year, so we'll go by calendar year instead.*

```
res1 = mutate(res1, Month = month(Date), Year = year(Date)) %>%  
  filter(Region != "SDWS") %>%  
  left_join(Yrs)
```

```
## Joining with `by = join_by(Year)`
```

```
ggplot(res1, aes(x = Location, y = ResTime)) + geom_boxplot()+  
  facet_grid(Yr_type~Region, scales = "free")
```



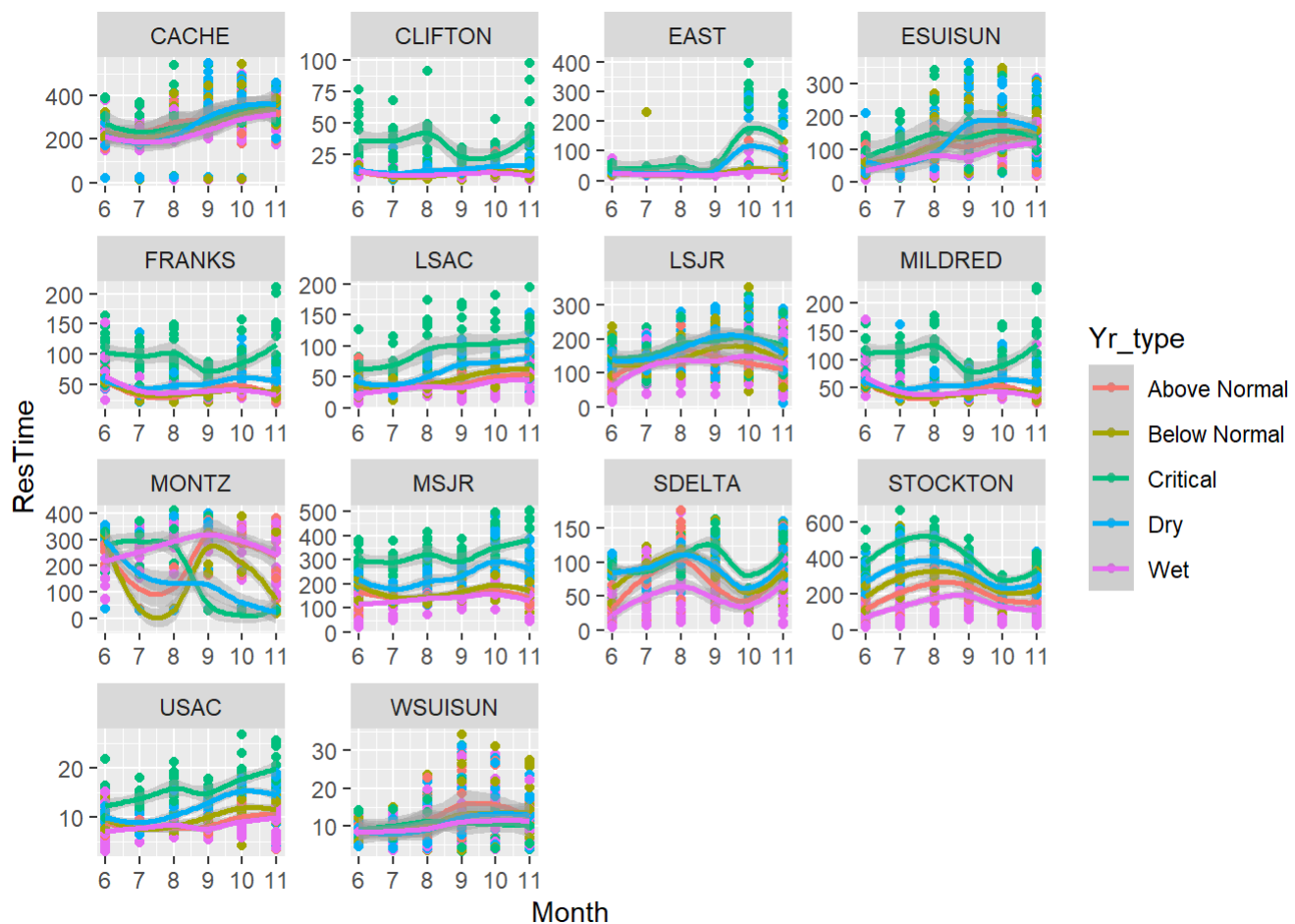
now let's calculate averages by region

```
resave = group_by(res1, SimPeriod, Region, Date, Yr_type, Month, Year, Index) %>%
  summarize(ResTime = mean(ResTime)) %>%
  mutate(DOY = yday(Date))
```

```
## `summarise()` has grouped output by 'SimPeriod', 'Region', 'Date', 'Yr_type',
## 'Month', 'Year'. You can override using the `.groups` argument.
```

```
ggplot(resave, aes(x = Month, y = ResTime, color = Yr_type)) + geom_point()+
  geom_smooth()+
  facet_wrap(~Region, scales = "free")
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



OK, now let's merge the microcystis data onto the regions. but we need to think about how to do it. We're trying to use microcystis from the whole time series, but it isn't exactly the same years as the residence time modeling. I think the most logical thing is to average by month and water year type.

```
HABregions15 = st_read("HABregions15.shp") %>%
  st_transform(crs = 4326) %>%
  st_make_valid()
```

```
## Reading layer `HABregions15' from data source
## `C:\Users\rhartman\OneDrive - California Department of Water Resources\HABs\HABsynthesis\HABregions15.shp'
## using driver `ESRI Shapefile'
## Simple feature collection with 15 features and 2 fields
## Geometry type: POLYGON
## Dimension: XY
## Bounding box: xmin: 574326.3 ymin: 4180347 xmax: 652771.2 ymax: 4272381
## Projected CRS: NAD83 / UTM zone 10N
```

```
#summarize residence time by water year type
resave2 = group_by(resave, Region, Yr_type, Month) %>%
  summarize(ResTime = mean(ResTime)) %>%
  rename(RegionDSM = Region)
```

```
## `summarise()` has grouped output by 'Region', 'Yr_type'. You can override using
## the `.groups` argument.
```

```
#switch the names so they match the other region names
reglook = read_csv("regionlookup.csv")
```

```
## Rows: 15 Columns: 2
## — Column specification —————
## Delimiter: ","
## chr (2): RegionDSM, Region
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
resave2 = left_join(resave2, reglook)
```

```
## Joining with `by = join_by(RegionDSM)`
```

```
#Load HABs data
load("data/HABs.RData")

#Make the data frame a sf object and join it to the regions
HABsReg = st_as_sf(filter(HABs, !is.na(Latitude)), coords = c("Longitude", "Latitude"), crs = 4326) %>%
  st_join(HABregions15) %>%
  mutate(Year = case_when(Source == "FMWTx" ~ 2021,
                          TRUE ~ Year)) %>%
  left_join(Yrs)
```

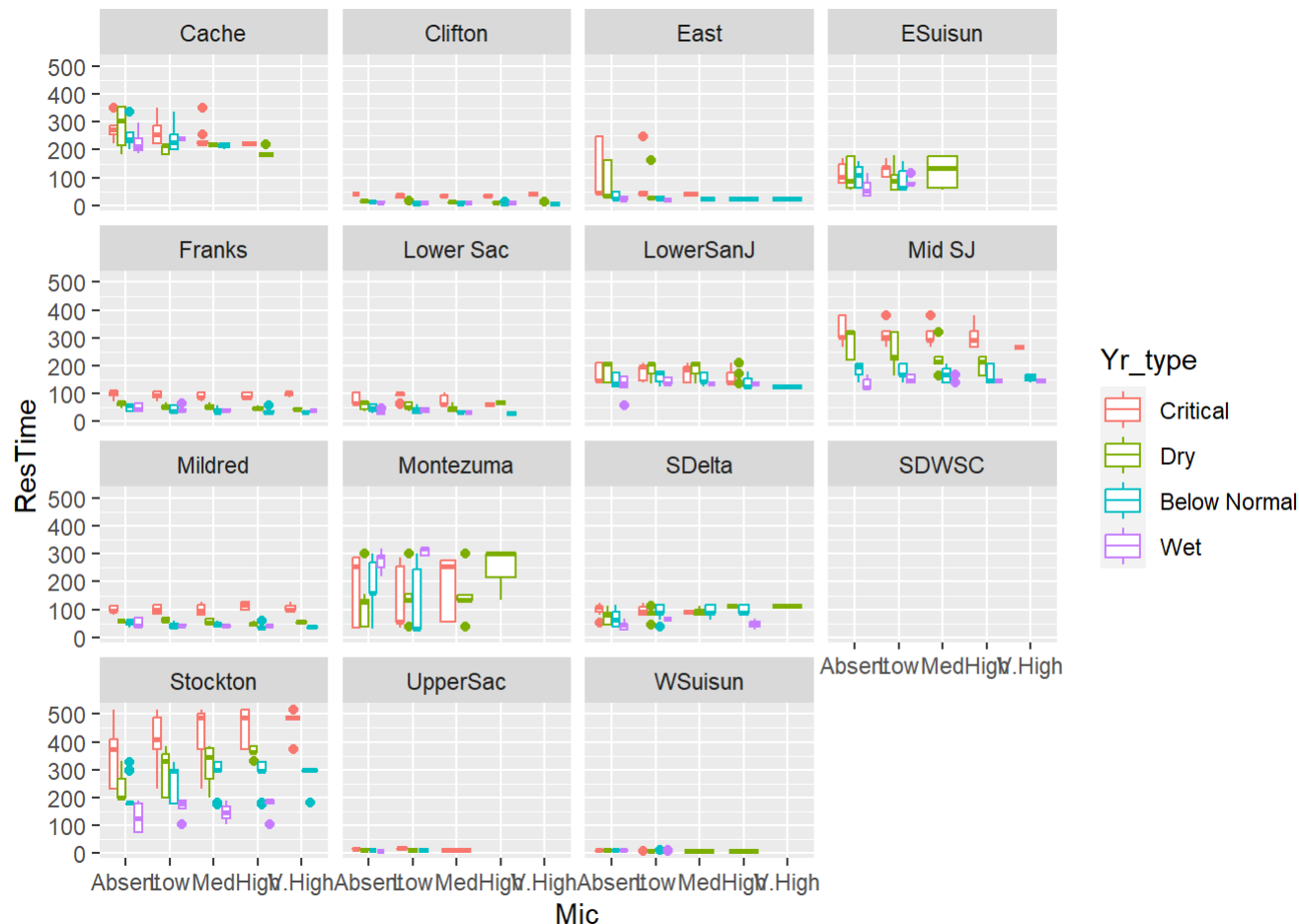
```
## Joining with `by = join_by(Year)`
```

```
#join restime and HABs by region and water year type
HABrestime = left_join(HABsReg, resave2) %>%
  filter(!is.na(Microcystis), Source != "DOP", !is.na(Region), Month %in% c(6,7,8,9,10)) %>%
  mutate(Mic = factor(round(Microcystis), levels = c(1,2,3,4,5), labels = c("Absent", "Low", "Med", "High", "V.High")),
         Yr_type = factor(Yr_type, levels = c("Critical", "Dry", "Below Normal", "Above Normal", "Wet")))
```

```
## Joining with `by = join_by(Month, Region, Yr_type)`
```

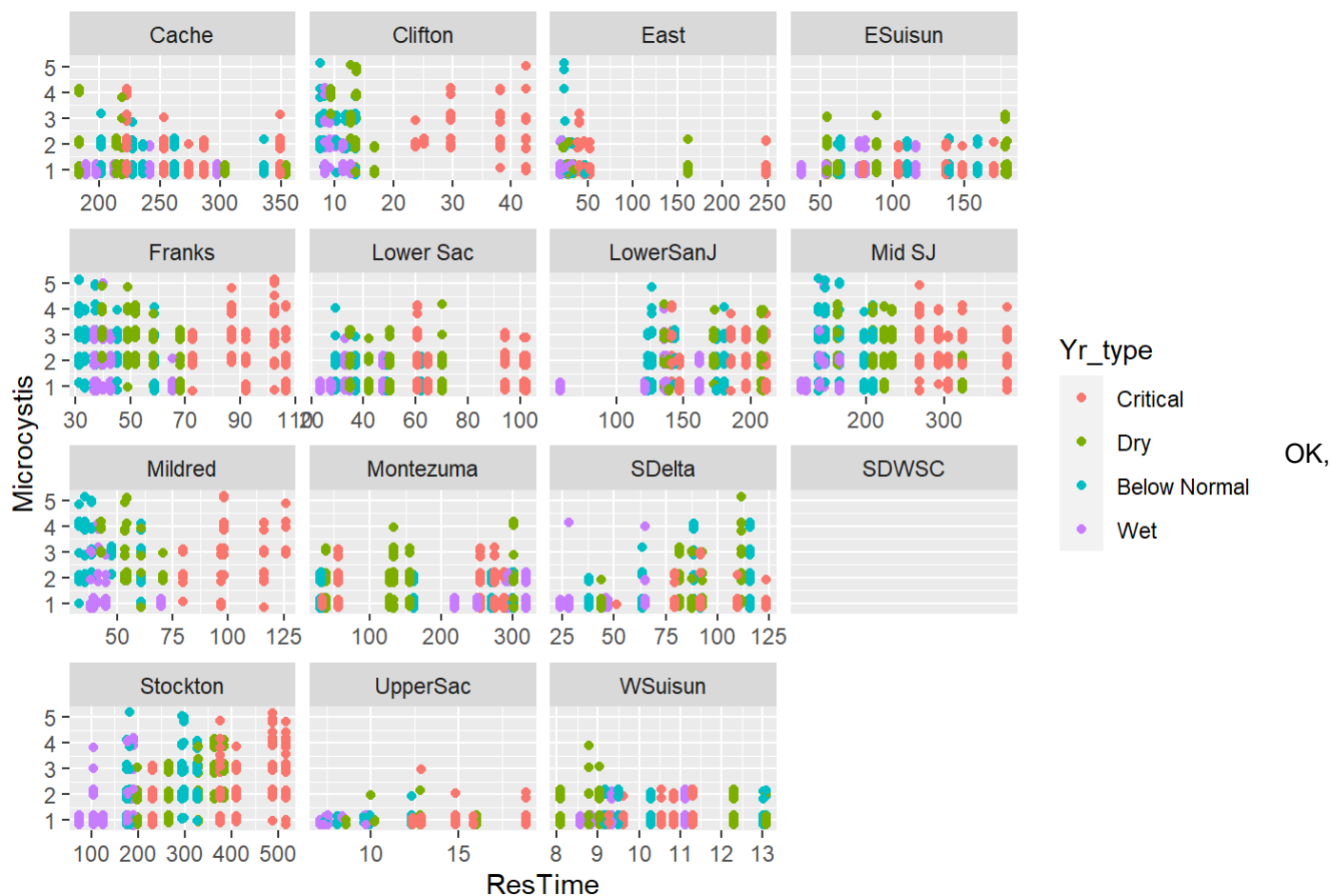
```
#boxplot of residence time by microcystis scores
ggplot(HABrestime, aes(x =Mic, y = ResTime, color = Yr_type)) +
  geom_boxplot()+
  facet_wrap(~Region)
```

```
## Warning: Removed 366 rows containing non-finite values (`stat_boxplot()`).
```



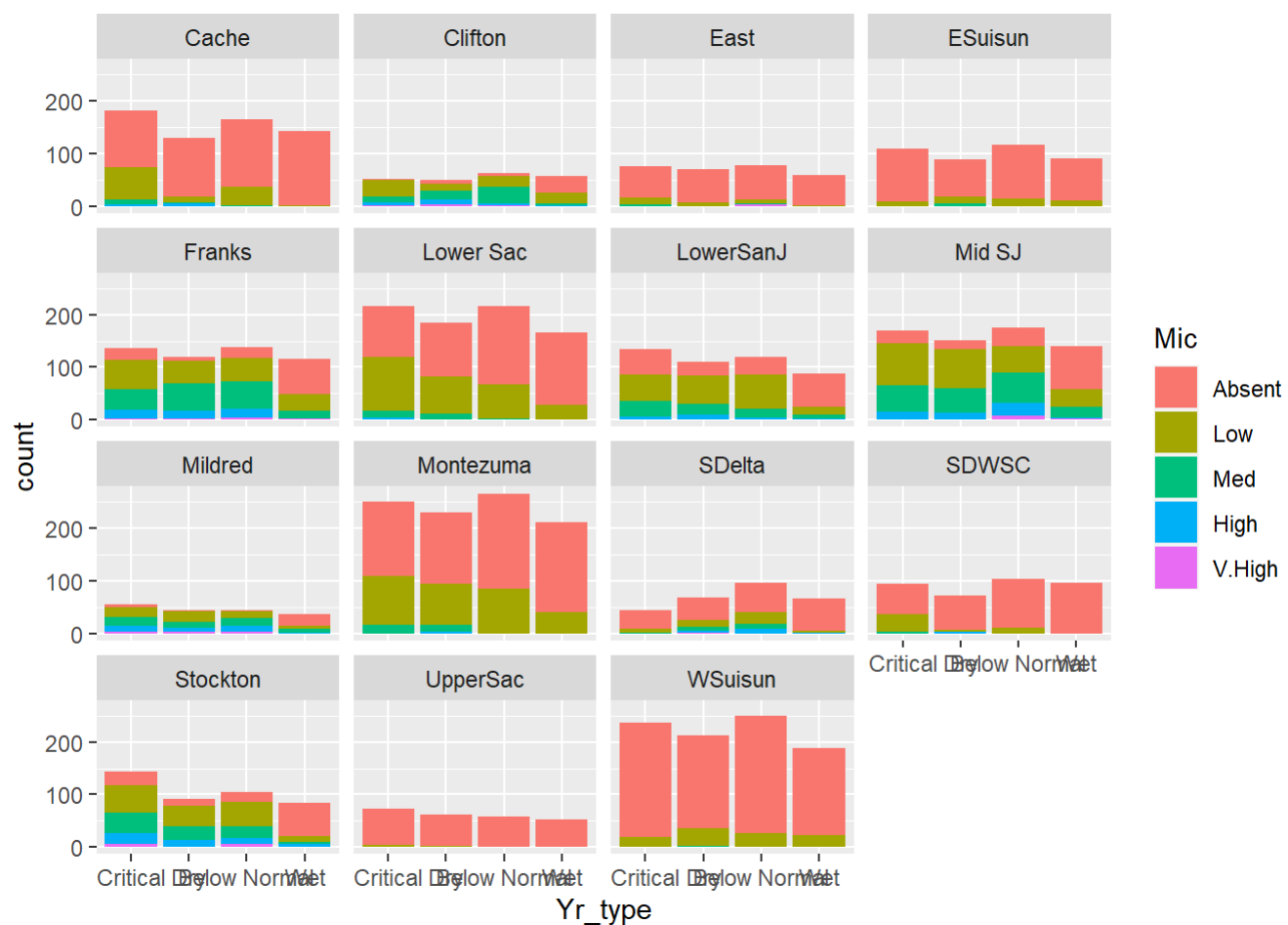
```
#scatterplot of microcystis scores by residence time
ggplot(HABrestime, aes(x =ResTime, y = Microcystis, color = Yr_type)) +
  geom_point(position = "jitter")+
  facet_wrap(~Region, scales = "free_x")
```

```
## Warning: Removed 366 rows containing missing values (`geom_point()`).
```

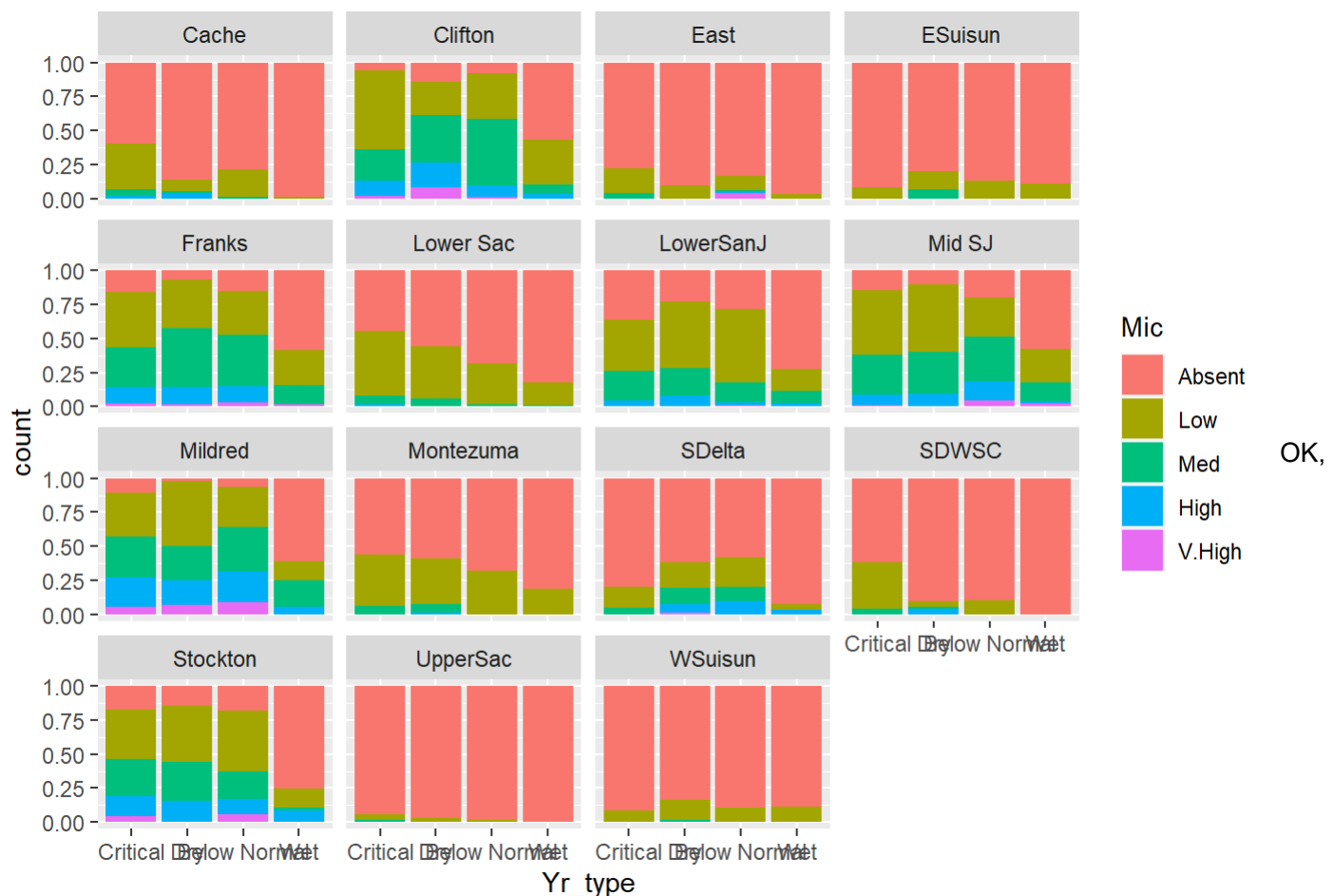


not let's try everything together/

```
#bar plot of frequency of microcystis by water year type and region
ggplot(HABrestime, aes(x =Yr_type, fill = Mic)) +
  geom_bar()+
  facet_wrap(~Region)
```

```
#percentage version
ggplot(HABrestime, aes(x =Yr_type, fill = Mic)) +
  geom_bar(position = "fill")+
  facet_wrap(~Region)
```



we see that drier year types have more Microcystis than wetter year types. Also a lot of variation across regions, with the central and south delta having the most

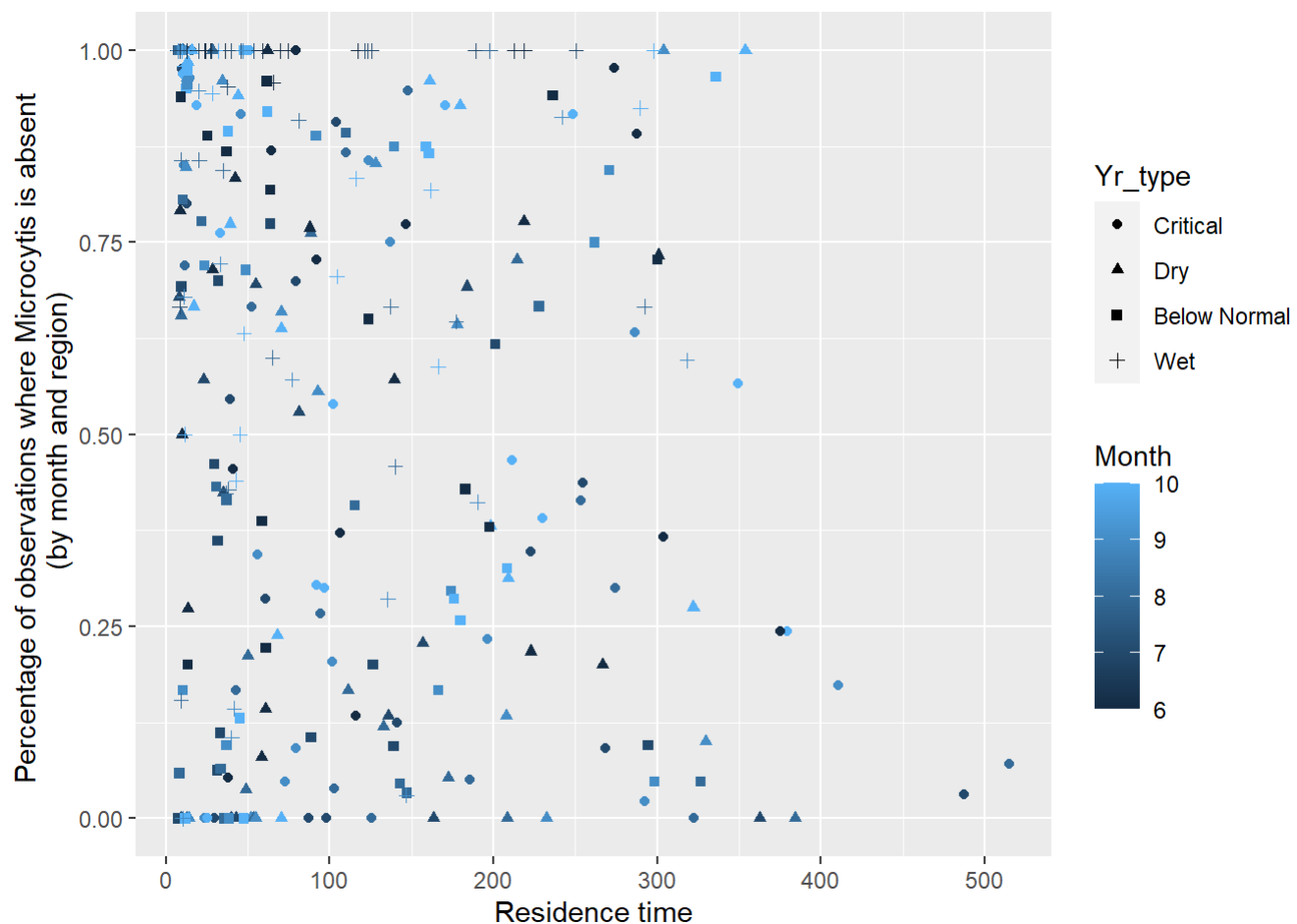
```
#now maybe % microcystis by water year type?
```

```
HABRT2 = group_by(HABrestime, Yr_type, Region, Month) %>%
  summarise(Restime = mean(ResTime), n = n(), Mic0 = length(which(Mic == "Absent"))/n,
    MicLow = length(which(Mic %in% c("Low", "Med")))/n,
    MicHigh = length(which(Mic %in% c("High", "V.High")))/n)
```

```
## `summarise()` has grouped output by 'Yr_type', 'Region'. You can override using
## the `.groups` argument.
```

```
#this is frequency of absence
ggplot(HABRT2, aes(x = Restime, y = Mic0, color = Month, shape = Yr_type))+
  geom_point()+ ylab("Percentage of observations where Microcystis is absent \n (by month and region)")+
  xlab("Residence time")
)
```

```
## Warning: Removed 20 rows containing missing values (`geom_point()`).
```

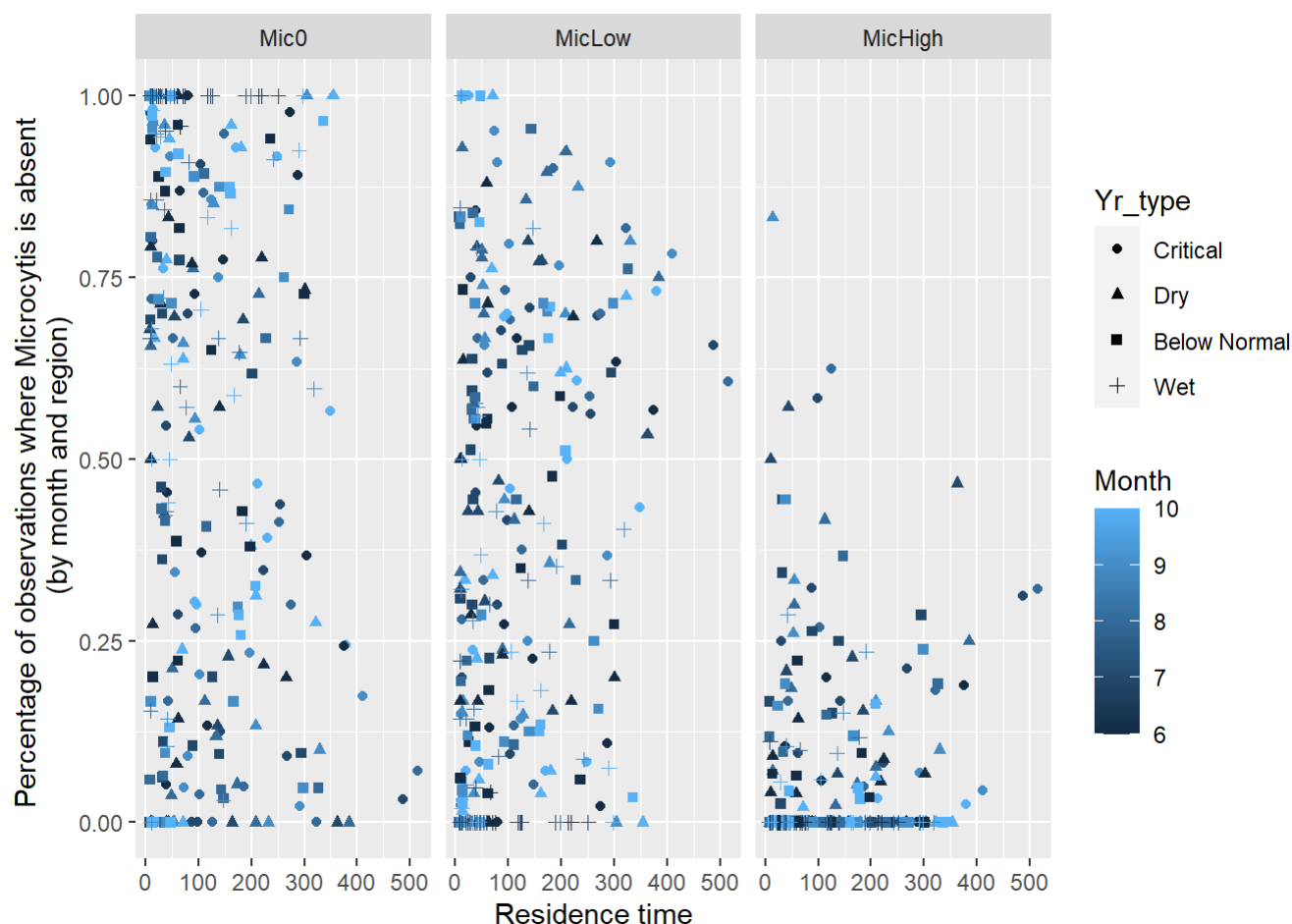


We can see that most of the months that have 100% absence are in wet years. There aren't as many patterns with month of the year as I was expecting, but it's a bit hard to see. Also, the longest residence time months/regions have a lot of microcystis.

```
#plot absence/presence/Low all together?
HABRT3 = pivot_longer(HABRT2, cols = c(Mic0, MicLow, MicHigh), names_to = "MicLevel", values_to = "Percentage") %>%
  mutate(MicLevel = factor(MicLevel, levels = c("Mic0", "MicLow", "MicHigh")))

ggplot(HABRT3, aes(x = Restime, y = Percentage, color = Month, shape = Yr_type))+
  geom_point()+ ylab("Percentage of observations where Microcystis is absent \n (by month and region)")+
  xlab("Residence time")
)+
  facet_wrap(~MicLevel)
```

```
## Warning: Removed 60 rows containing missing values (`geom_point()`).
```



OK, so maybe there is a pattern here? It's honestly really hard to tell.

```
library(ordinal)
```

```
##
## Attaching package: 'ordinal'
```

```
## The following object is masked from 'package:dplyr':
##
## slice
```

```
library(effects)
```

```
## Loading required package: carData
```

```
## lattice theme set by effectsTheme()
## See ?effectsTheme for details.
```

```
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##      select
```

```
#Let's try a model of HABs versus temperature, and residence time
HABrestime = mutate(HABrestime, MicF = case_when(Microcystis ==1 ~ "Absent",
                                                  Microcystis %in% 2:3~ "Mid",
                                                  Microcystis %in% 4:5 ~ "High"),
                   MicF = factor(MicF, levels = c("Absent", "Mid", "High"), ordered = T))

#unfortunately, we are missing a lot of temperature values
HABtest = filter(HABrestime,!is.na(Temperature), !is.na(ResTime), !is.na(Station), !is.na(MicF))

m1 = clmm(MicF ~ ResTime+ Temperature+ (1|Station)+ (1|Year), data = HABtest)

summary(m1)
```

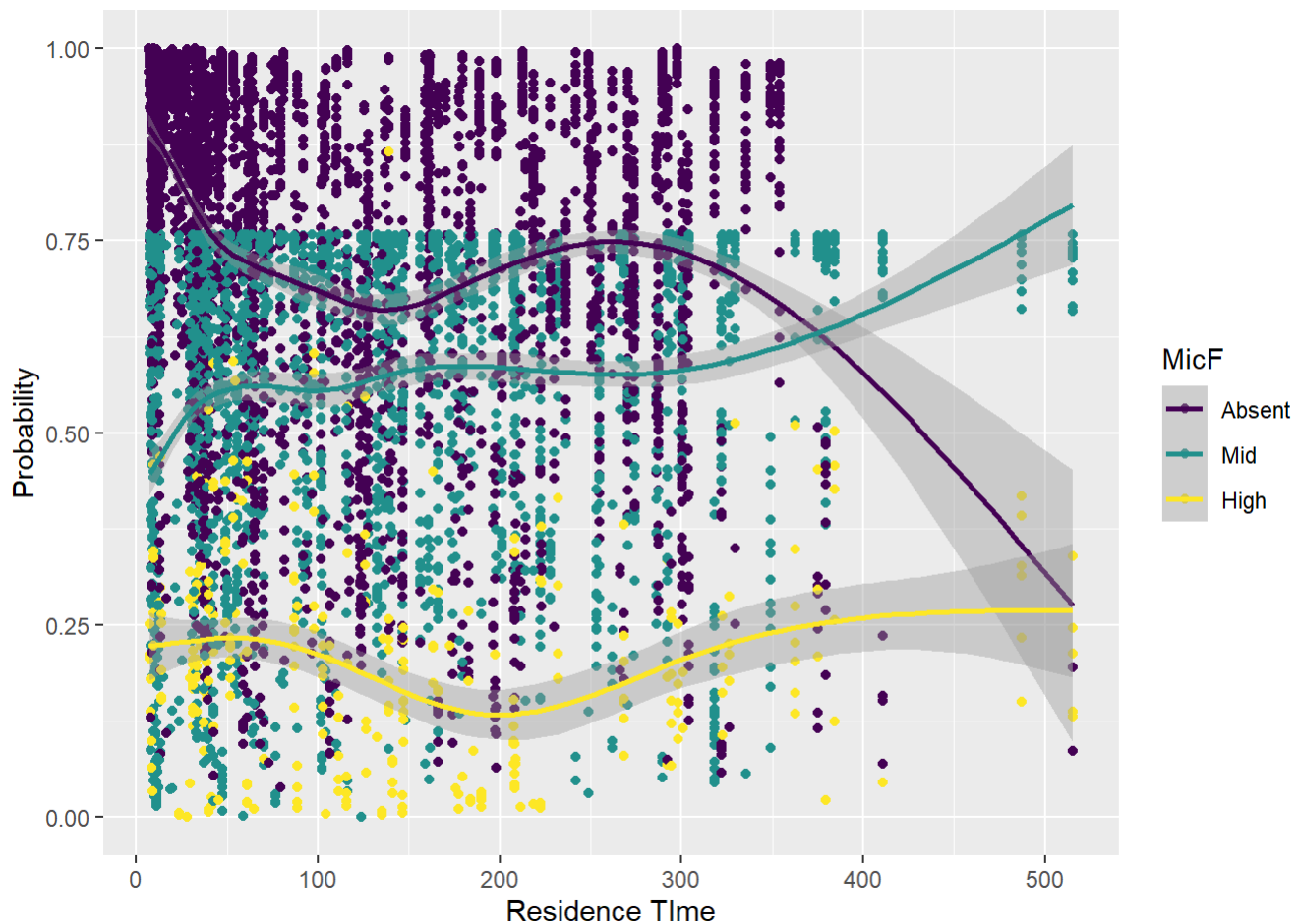
```
## Cumulative Link Mixed Model fitted with the Laplace approximation
##
## formula: MicF ~ ResTime + Temperature + (1 | Station) + (1 | Year)
## data:    HABtest
##
## link threshold nobs logLik  AIC      niter    max.grad cond.H
## logit flexible 6819 -4001.74 8015.49 802(4007) 7.65e-02 2.1e+06
##
## Random effects:
## Groups Name      Variance Std.Dev.
## Station (Intercept) 2.4752  1.5733
## Year (Intercept) 0.7954  0.8919
## Number of groups: Station 168, Year 16
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## ResTime      0.0005175  0.0004900  1.056    0.291
## Temperature  0.3474109  0.0183761 18.906 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Threshold coefficients:
##              Estimate Std. Error z value
## Absent|Mid    8.1464    0.4723 17.25
## Mid|High     12.1152    0.4956 24.45
```

```
#not as exciting as I'd think
#Also, i need to work on how to plot this.
```

```
HABtest$fits = fitted(m1)

ggplot(HABtest, aes( ResTime, fits, color = MicF))+ geom_point()+ geom_smooth()+
  ylab("Probability")+ xlab("Residence TIme")
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



```
#well, that's a mess
```

Dry years only

```
#maybe if we standardized by water year type in some way?
#Let's try just dry and critical years
```

```
dry = filter(HABtest, Yr_type %in% c("Dry", "Critical"))

m1 = clmm(MicF ~ ResTime+ Temperature+ (1|Station)+ (1|Year), data = dry)

summary(m1)
```

```
## Cumulative Link Mixed Model fitted with the Laplace approximation
##
## formula: MicF ~ ResTime + Temperature + (1 | Station) + (1 | Year)
## data:    dry
##
## link threshold nobs logLik  AIC      niter      max.grad cond.H
## logit flexible  3429 -2066.79 4145.59 293(1420) 1.30e-02 1.8e+06
##
## Random effects:
## Groups Name          Variance Std.Dev.
## Station (Intercept) 3.0735    1.753
## Year (Intercept) 0.2652    0.515
## Number of groups:  Station 163, Year 9
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## ResTime      0.0007787  0.0007769   1.002    0.316
## Temperature  0.4725691  0.0271305  17.418 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Threshold coefficients:
##              Estimate Std. Error z value
## Absent|Mid  10.6653    0.6332   16.84
## Mid|High    15.2456    0.6852   22.25
```

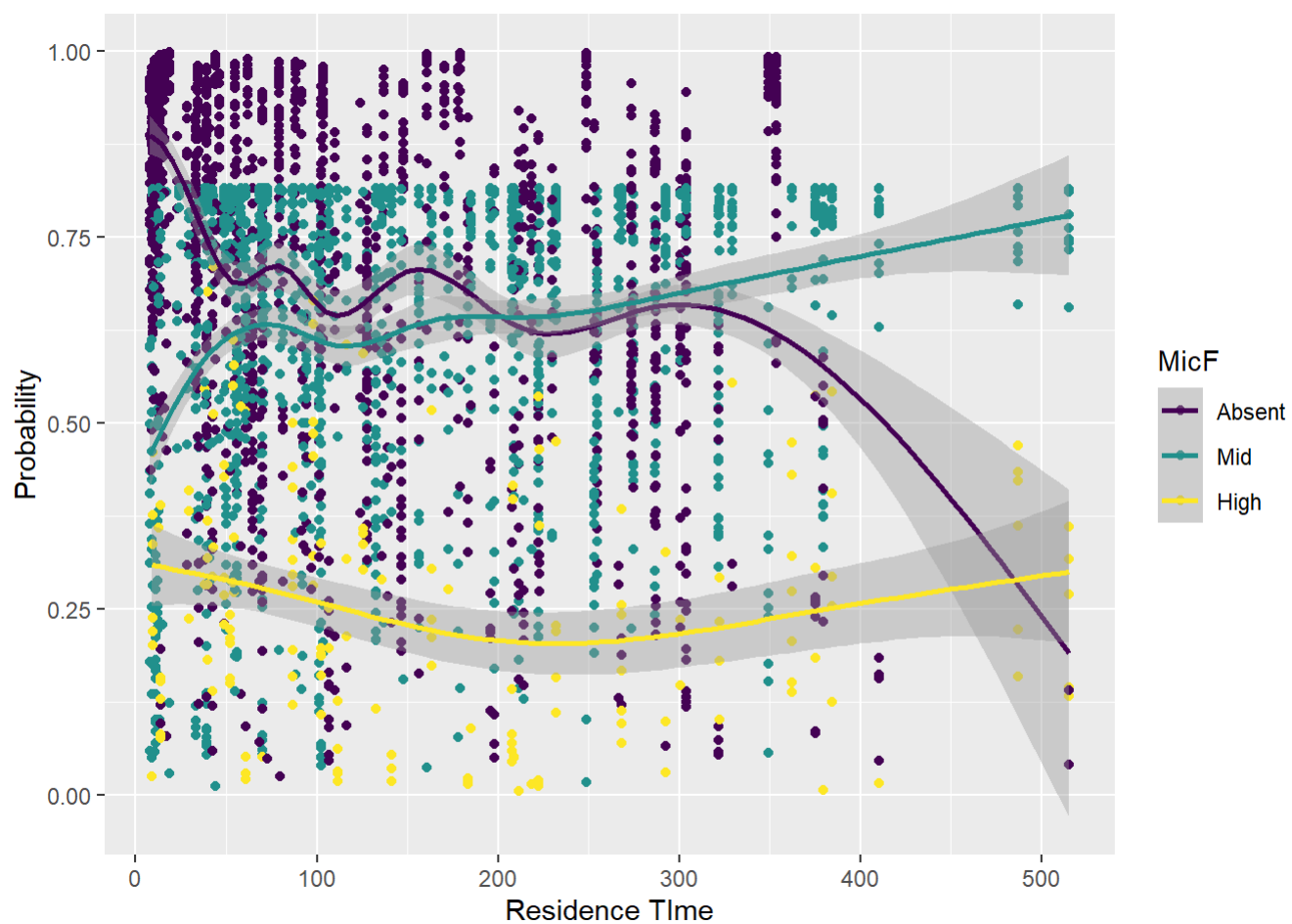
#not as exciting as I'd think

#Also, i need to work on how to plot this.

```
dry$fits = fitted(m1)
```

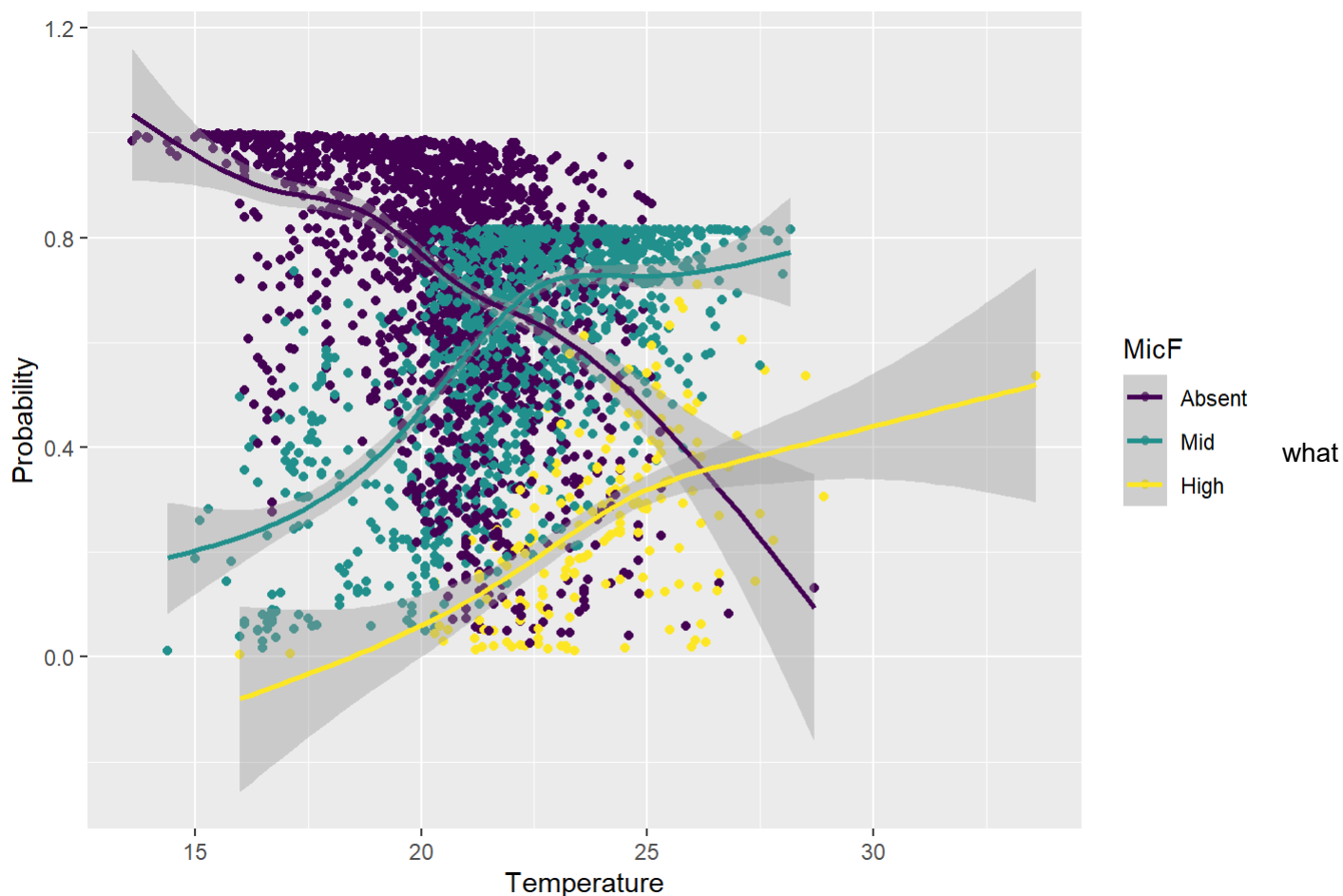
```
ggplot(dry, aes( ResTime, fits, color = MicF))+ geom_point()+ geom_smooth()+
  ylab("Probability")+ xlab("Residence TIme")
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



```
ggplot(dry, aes(Temperature, fits, color = MicF))+ geom_point()+ geom_smooth()+
  ylab("Probability")+ xlab("Temperature")
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```

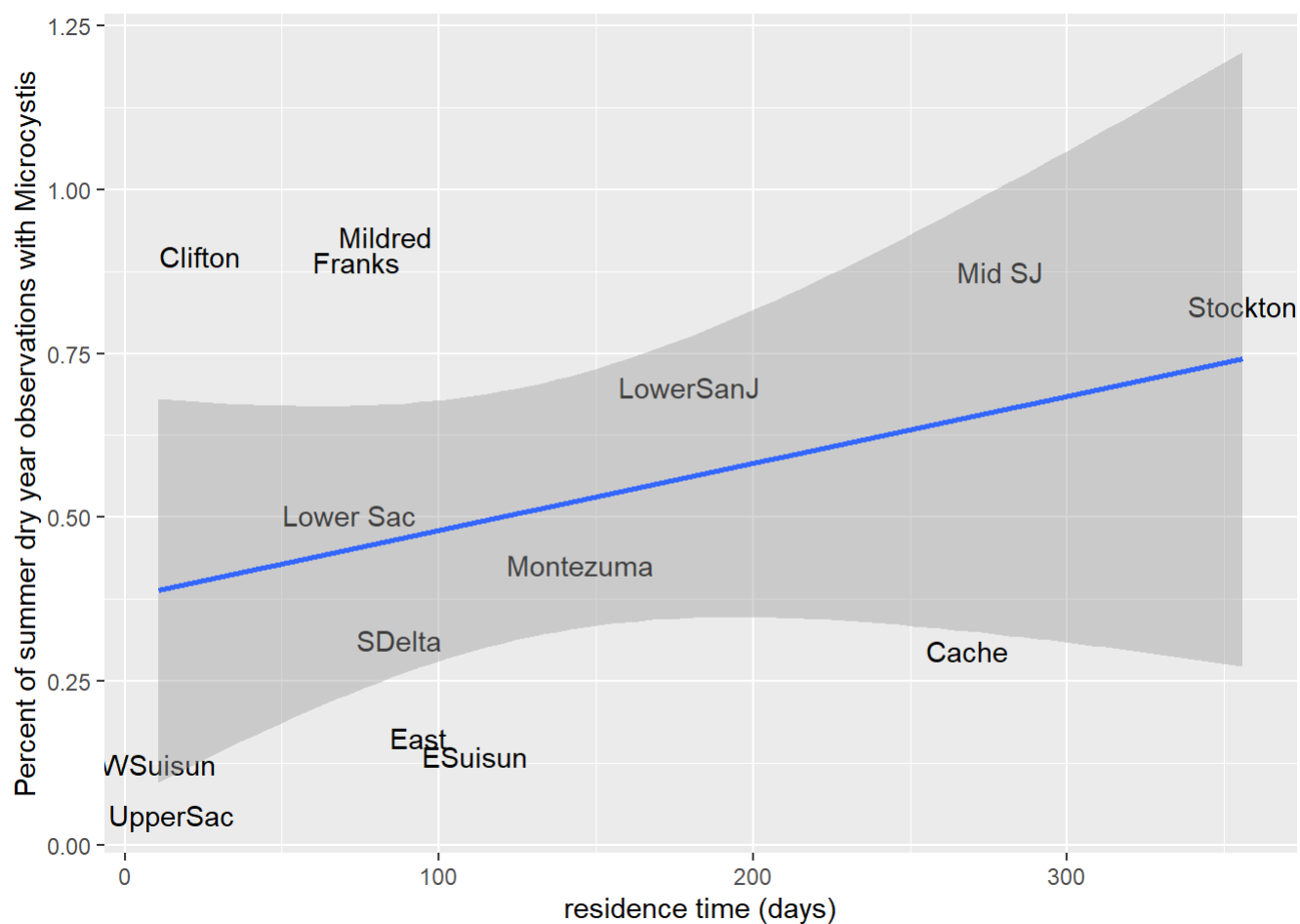



if we did a binomial model by region?

```
micbin = group_by(dry, Region) %>%
  summarize(Present = length(Microcystis[which(Microcystis>1)]), Absent = length(Microcystis[whi
ch(Microcystis==1)]),
            ResTime = mean(ResTime), Temperature = mean(Temperature))

ggplot(micbin, aes(x = ResTime, y = Present/(Present+Absent))) + geom_text(aes(label = Region))+
  ylab("Percent of summer dry year observations with Microcystis")+ xlab("residence time (day
s)") +
  geom_smooth(method = "lm")
```

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

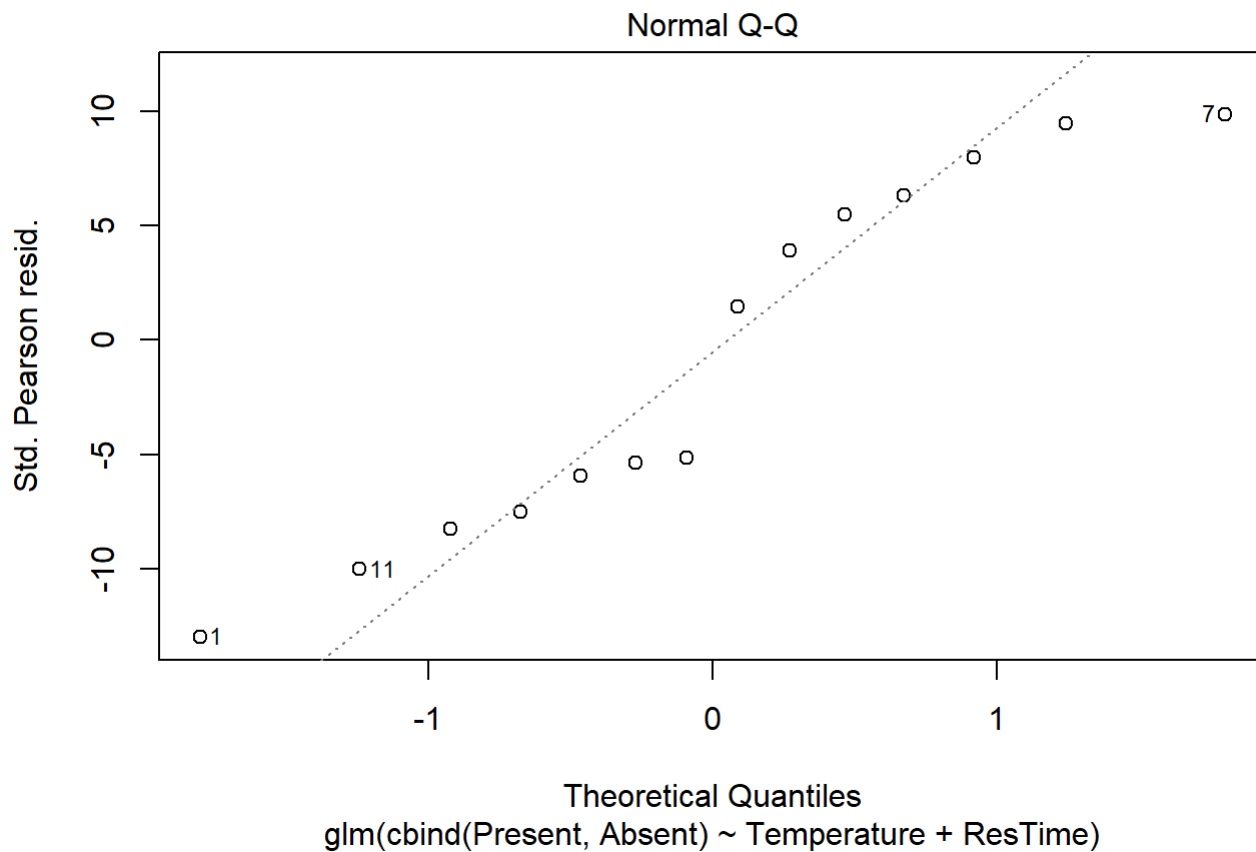
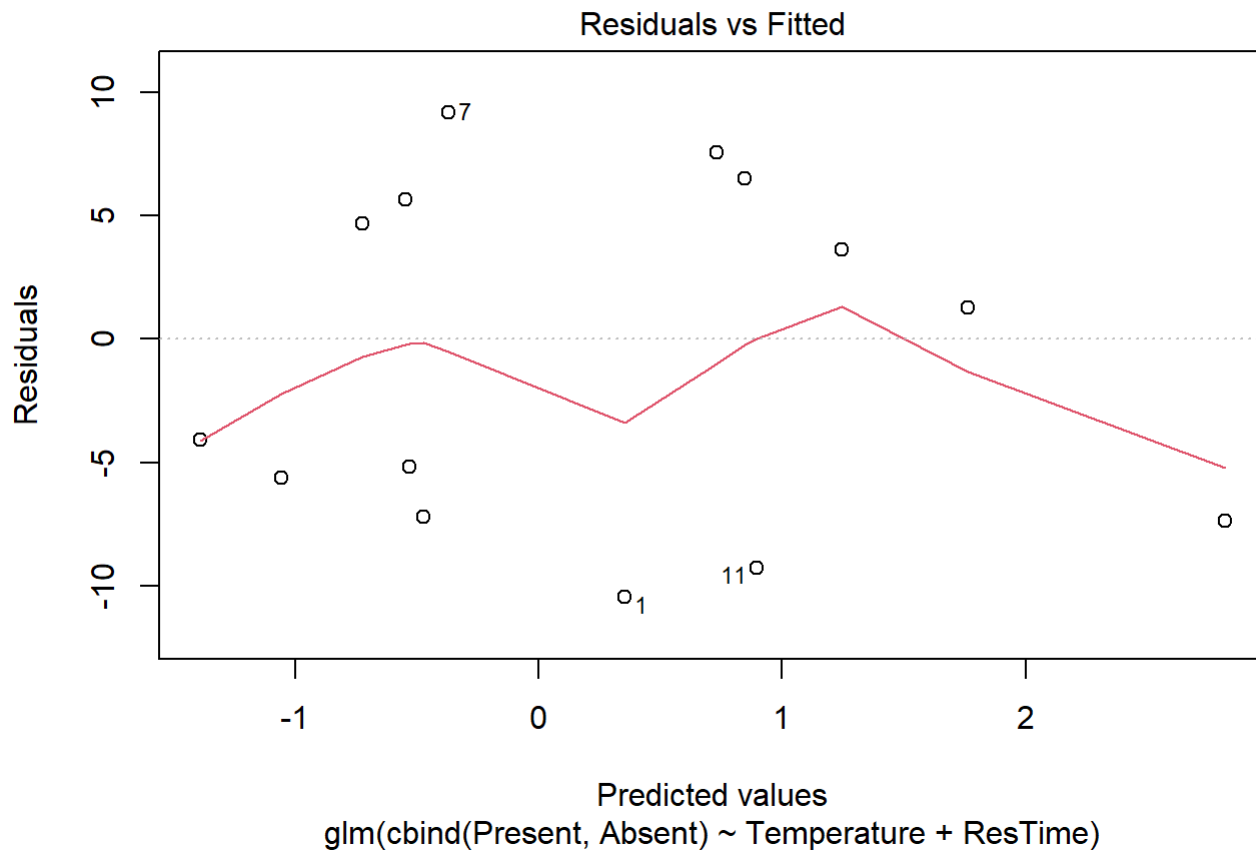


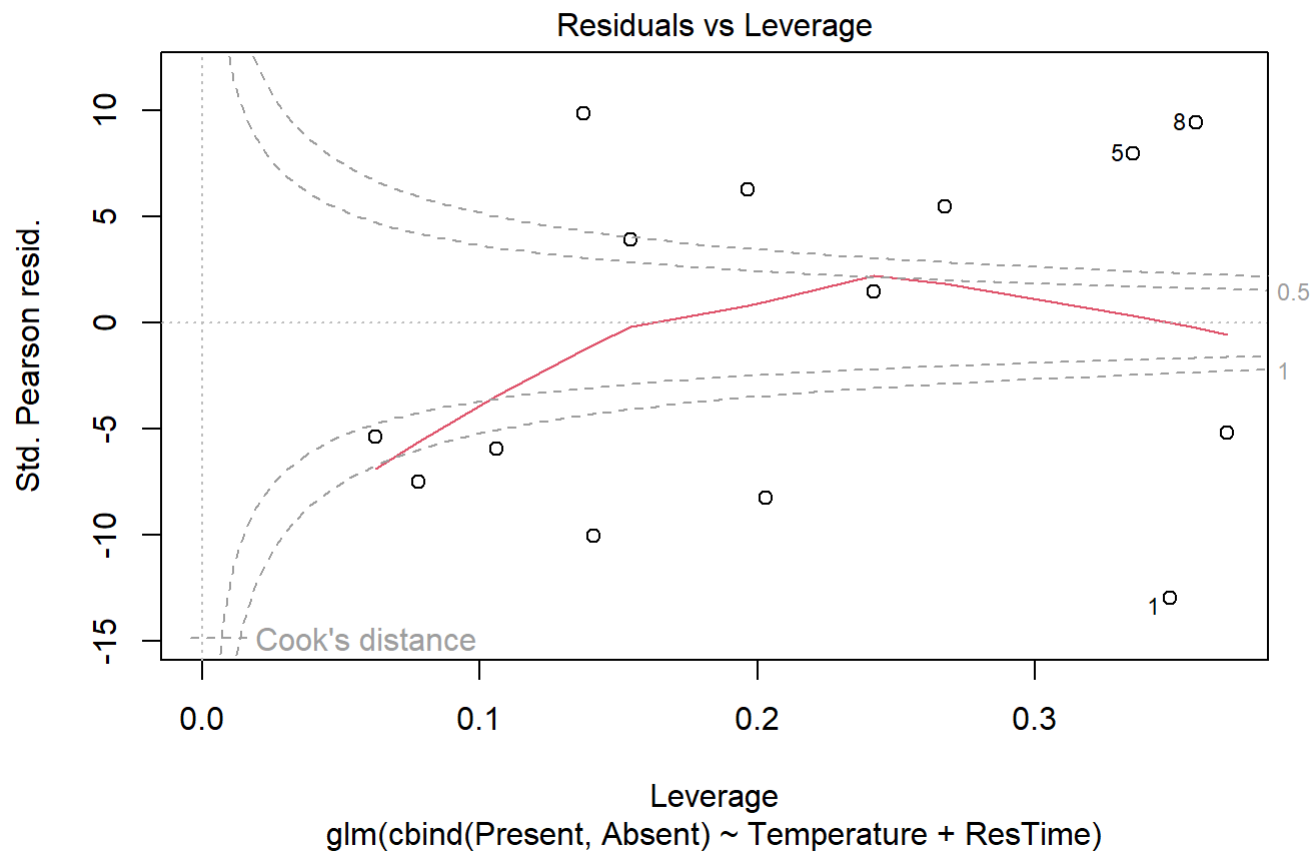
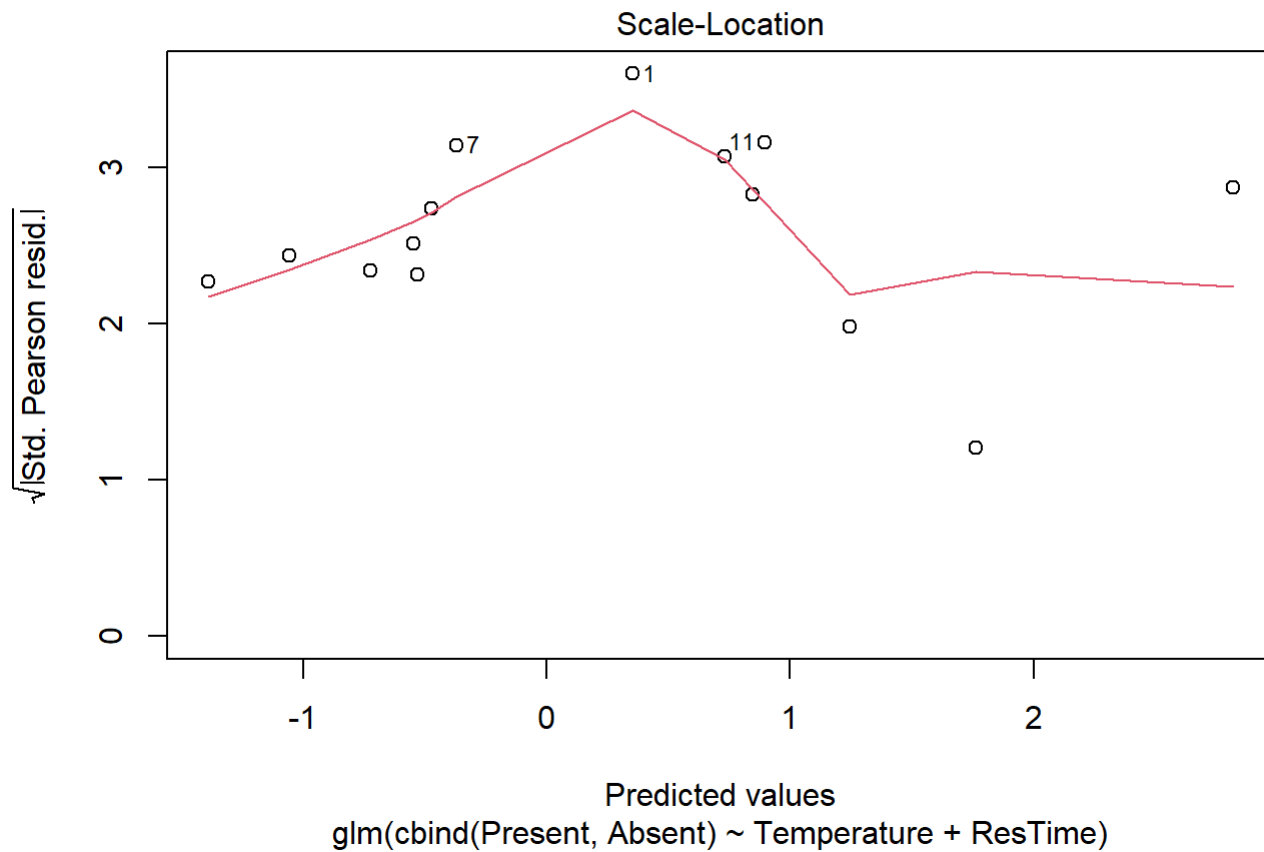
don't hate this.

```
bn1 = glm(cbind(Present, Absent)~ Temperature + ResTime, data = micbin, family = "binomial")
summary(bn1)
```

```
##
## Call:
## glm(formula = cbind(Present, Absent) ~ Temperature + ResTime,
##      family = "binomial", data = micbin)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -10.411   -6.425   -1.498    5.313    9.102
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.785e+01  8.844e-01 -20.185  < 2e-16 ***
## Temperature  8.193e-01  4.238e-02  19.332  < 2e-16 ***
## ResTime      2.816e-03  4.295e-04   6.556 5.53e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 1348.00  on 13  degrees of freedom
## Residual deviance:  644.16  on 11  degrees of freedom
## AIC: 722.58
##
## Number of Fisher Scoring iterations: 5
```

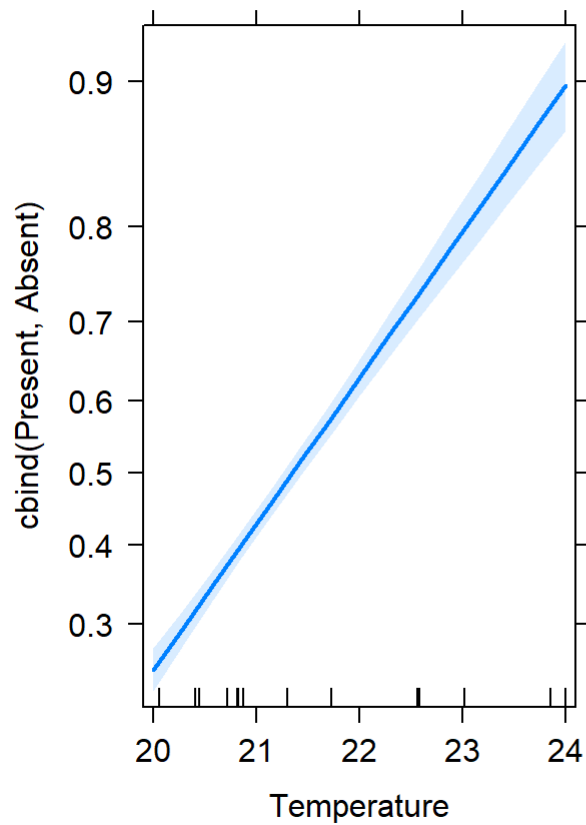
```
plot(bn1)
```





```
#OK, not too bad
```

```
library(effects)  
plot(allEffects(bn1))
```

Temperature effect plot**ResTime effect plot**