Customise a ggplot for a dataset with interactions between variables

JR Ferrer-Paris (https://github.com/jrfep)

2021-06-21

I was given a dataset and a glm model with interactions, and the user wants to show several interactions in a single plot

Libraries

First load the libraries

```
require(ggplot2)
```

```
## Loading required package: ggplot2
```

```
require(ggnewscale)
```

```
## Loading required package: ggnewscale
```

Dataset

```
n.species <- c(20, 12, 8, 15, 9, 15, 15, 8, 6, 3, 12, 9, 19, 3, 2, 36, 13, 27, 13, 16, 15, 7, 12, 36, 3
##n.species <- c(4, 4, 9, 7, 7, 6, 11, 7, 18, 16, 9, 14, 2, 14, 15, 5, 3, 11, 4, 9, 8, 7, 6, 8, 7, 4, 4
log.area <- c(4.600786168, 4.600786168, 2.587377991, 0.690311286, 0.123721006, 1.475014166, 0.93399822,
rain.diff <- c(98.3, 111.6, -172.8, 102.1, -126.8, -126.8, -29.8, 64.6, 72.5, 133.4, 66.9, 155.5, -119.
blocks <- factor(c(2, 2, 2, 2, 1, 1, 1, 1, 3, 3, 1, 2, 3, 3, 3, 1, 1, 3, 3, 1, 1, 1, 2, 1, 1, 1, 1, 2,

obs.data <- data.frame(n.species,log.area,rain.diff,blocks)
```

Model

This model has many interactions, not all are important, but we want to visualise the differences:

```
mdl <- glm(n.species ~ (log.area*blocks) * (rain.diff*log.area), data=obs.data, family=quasipoisson())
summary(mdl)
```

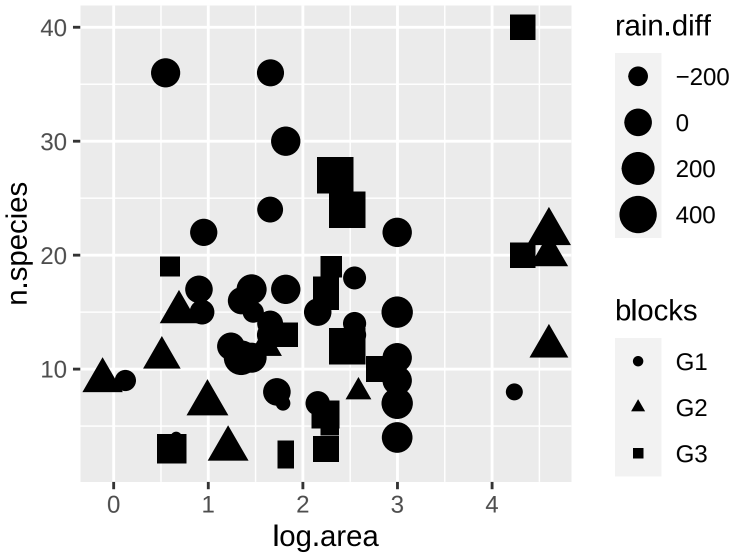
```
##
## Call:
## glm(formula = n.species ~ (log.area * blocks) * (rain.diff *
##     log.area), family = quasipoisson(), data = obs.data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.2061  -1.5074  -0.2677   1.1821   4.3301
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    2.9839559  0.2076406  14.371  < 2e-16 ***
## log.area                    -0.1542033  0.1027541  -1.501  0.13998
## blocksG2                    -0.4407950  0.6394708  -0.689  0.49395
## blocksG3                    -1.6743844  0.5209803  -3.214  0.00234 **
## rain.diff                    0.0026900  0.0013127   2.049  0.04592 *
## log.area:blocksG2            0.1057303  0.2551635   0.414  0.68045
## log.area:blocksG3            0.6447873  0.1898437   3.396  0.00138 **
## log.area:rain.diff          -0.0011579  0.0006023  -1.923  0.06048 .
## blocksG2:rain.diff          -0.0059650  0.0045429  -1.313  0.19542
## blocksG3:rain.diff          -0.0090611  0.0050920  -1.779  0.08149 .
## log.area:blocksG2:rain.diff  0.0025297  0.0015435   1.639  0.10776
## log.area:blocksG3:rain.diff  0.0043035  0.0021992   1.957  0.05620 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 3.882222)
##
##     Null deviance: 282.79  on 59  degrees of freedom
## Residual deviance: 186.37  on 48  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 5
```

Plot

We have four variables that we want to display, three are continuous, one categorical. If we use the two axes for two variables, how do we show the other two?

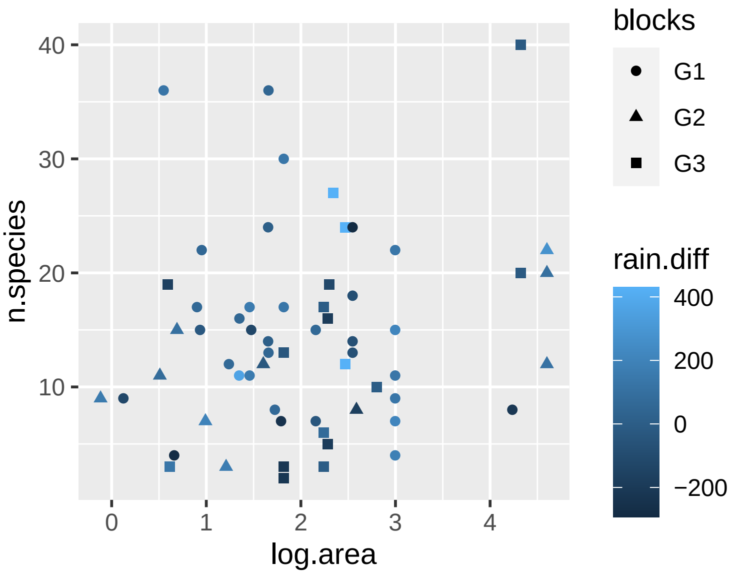
Size and shape:

```
ggplot(data=obs.data) +
  geom_point(aes(x=log.area,y=n.species, size=rain.diff, shape=blocks))
```



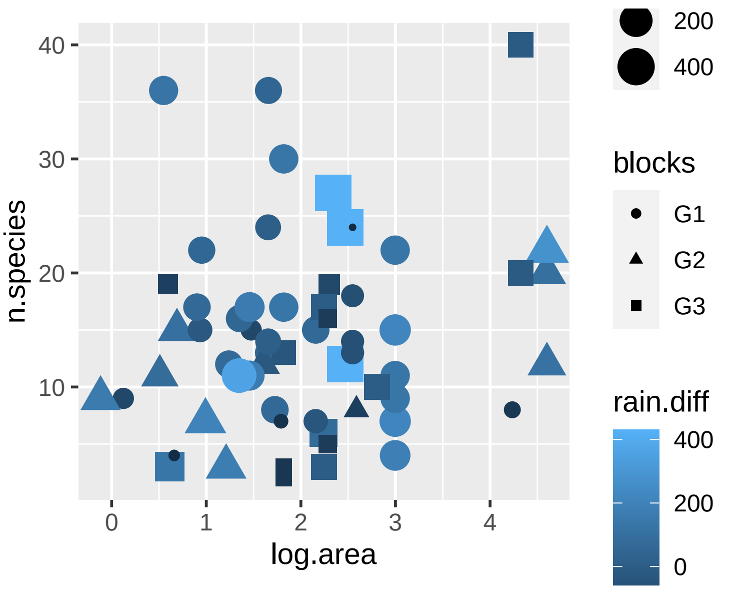
Colour and shape:

```
ggplot(data=obs.data) +
  geom_point(aes(x=log.area,y=n.species, colour=rain.diff, shape=blocks))
```



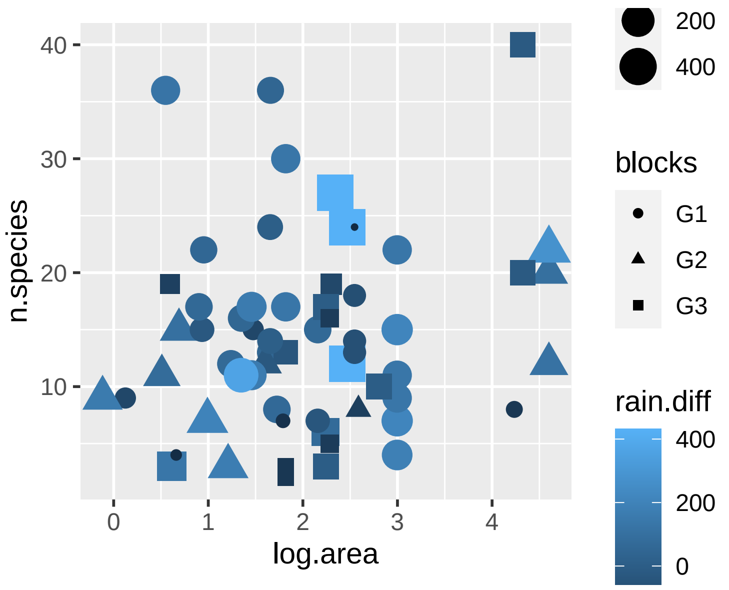
Or better size and colour and shape:

```
ggplot(data=obs.data) +
  geom_point(aes(x=log.area,y=n.species, colour=rain.diff, size=rain.diff,shape=blocks))
```



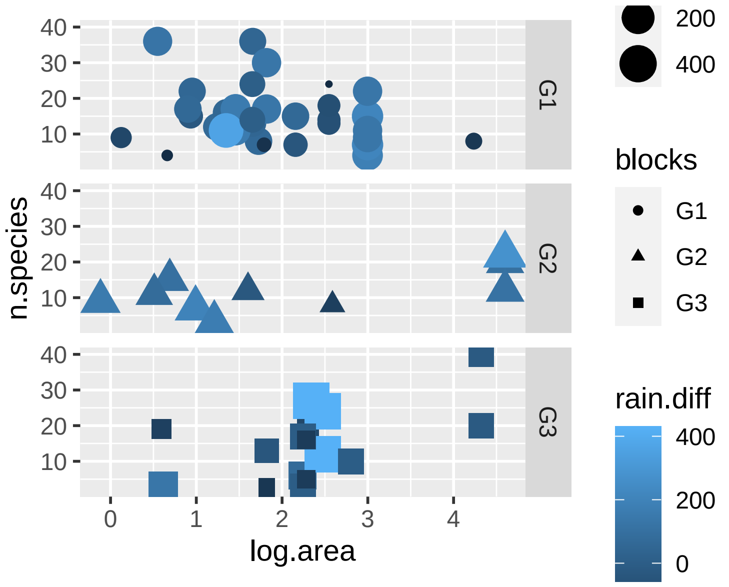
That is not so bad, but we want to improve on this, you can save the plot and add elements to it:

```
(plot_with4vars <- ggplot(data=obs.data) +
  geom_point(aes(x=log.area,y=n.species, colour=rain.diff, size=rain.diff,shape=blocks)))
```



The previous plot is still too cluttered, we can divide it in facets:

```
plot_with4vars + facet_grid(blocks~.)
```



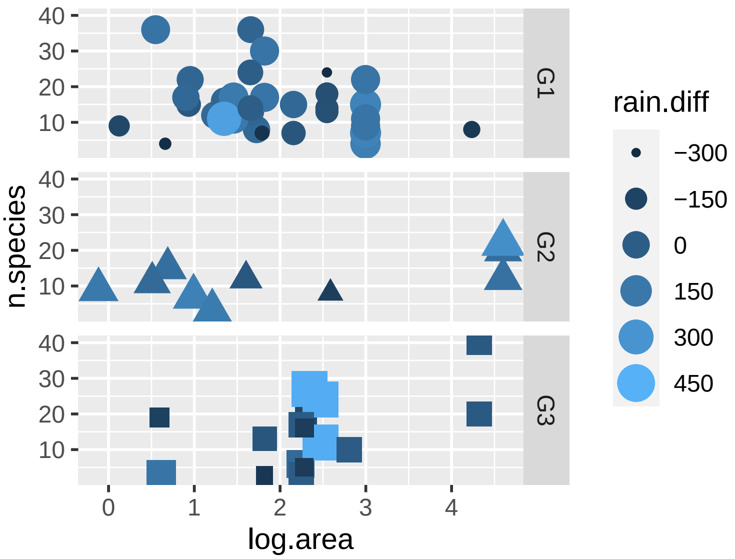
This is more readable, but we want to join the legends of size and colour, and maybe we don't need the legend for blocks, since we already have them in different panels:

```
range(obs.data$rain.diff)
```

```
## [1] -293.1  430.6
```

```
var3.limits <- c(-300,450)
var3.breaks <- seq(-300,450,by=150)

plot_with4vars + facet_grid(blocks~.) +
scale_size_continuous(limits=var3.limits, breaks=var3.breaks) +
scale_colour_continuous(limits=var3.limits, breaks=var3.breaks) +
guides(size=guide_legend(), colour=guide_legend(), shape='none')
```



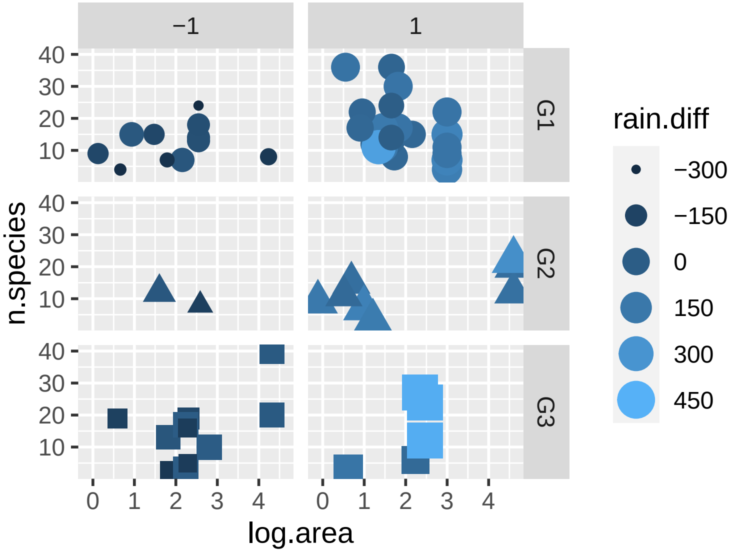
We further split the plot by the sign of the difference in rainfall, left size are the negative values, right side the positive values.

To do this we need to add a new variable to the data set and we better repeat the call to the original plot (otherwise we will get warnings and errors for missing faceting variables)

```
obs.data$rain.sign <- sign(obs.data$rain.diff)

plot_with4vars <- ggplot(data=obs.data) +
  geom_point(aes(x=log.area,y=n.species, colour=rain.diff, size=rain.diff,shape=blocks))

plot_with4vars + facet_grid(blocks~rain.sign) +
scale_size_continuous(limits=var3.limits, breaks=var3.breaks) +
scale_colour_continuous(limits=var3.limits, breaks=var3.breaks) +
guides(size=guide_legend(), colour=guide_legend(), shape='none')
```



Now we want to add the results of the model. We create a new data frame with new observations. We will use expand grid to create sequence of values along the range log.area and all levels of the blocks variables, but using only two quantiles of the rain.diff variable:

```
pred.data <- expand.grid(log.area=seq(min(obs.data$log.area), max(obs.data$log.area), length=20), rain.

pred.data$rain.sign <- factor(sign(pred.data$rain.diff))
```

We will add the predictions to this object. First predict the expected values and standard errors in the scale of the link function. Then, assuming assymptotic normality, we calculate the lower and upper intervals. Finally, we use the inverse link function to transform these values to the scale of the response variable.

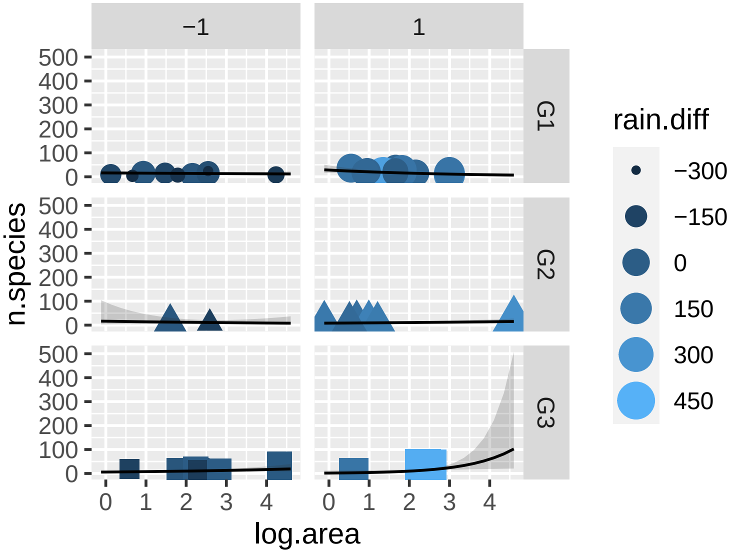
```
prd <- predict(mdl,newdata=pred.data,type='link',se.fit=T)
ilink <- family(mdl)$linkinv

pred.data$prediction <- ilink(prd$fit)
pred.data$pred.lower <- ilink(prd$fit-1.96*prd$se.fit)
pred.data$pred.upper <- ilink(prd$fit+1.96*prd$se.fit)
```

Now we can add the lines and a ribbon with the confidence intervals

```
plot_observed_data <- plot_with4vars + facet_grid(blocks~rain.sign) +
scale_size_continuous(limits=var3.limits, breaks=var3.breaks) +
scale_colour_continuous(limits=var3.limits, breaks=var3.breaks) +
guides(size=guide_legend(), colour=guide_legend(), shape='none')

plot_observed_data +
  geom_ribbon(data=pred.data, aes(x=log.area, ymin = pred.lower, ymax = pred.upper), alpha = 0.2, color
      geom_line(data=pred.data,aes(x=log.area,y=prediction) )
```

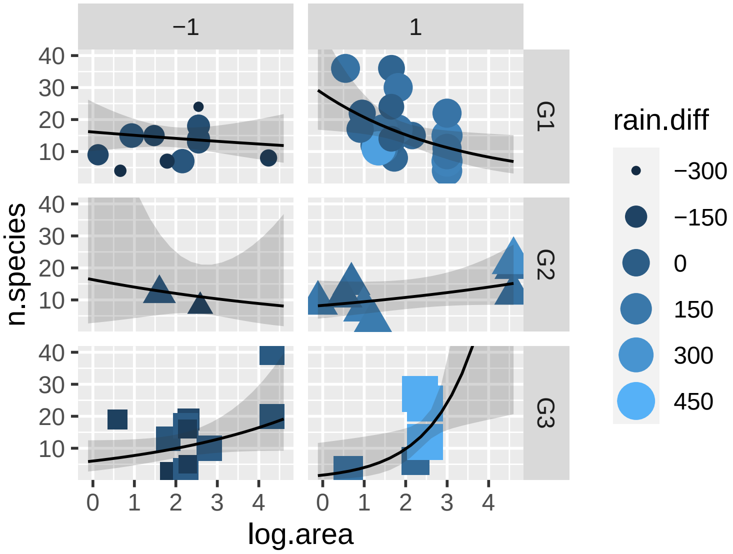


Well that looks… awful!

We have very large confidence intervals (I warned you, the model is not optimal). We need to limit our y-axis to the range of the observed values:

```
plot_obs_and_pred_data <- plot_observed_data +
  geom_ribbon(data=pred.data, aes(x=log.area, ymin = pred.lower, ymax = pred.upper), alpha = 0.2, color
      geom_line(data=pred.data,aes(x=log.area,y=prediction) )

plot_obs_and_pred_data + coord_cartesian(ylim = range(obs.data$n.species))
```



Well that looks better. It shows the estimated trend and the uncertainty of the model (very important!).

Maybe we can make this look even nicer with different colours of the predictions for negative and positive values of rain.diff:

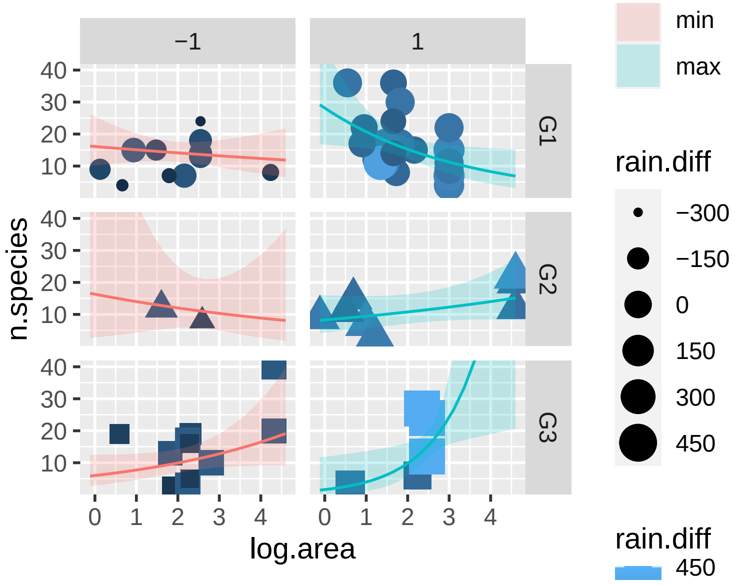
```
plot_observed_data +
  geom_ribbon(data=pred.data, aes(x=log.area, ymin = pred.lower, ymax = pred.upper, fill=rain.sign), al
      geom_line(data=pred.data,aes(x=log.area,y=prediction,colour=rain.sign) ) + coord_cartesian(ylim =
```

```
## Error: Discrete value supplied to continuous scale
```

Upps! that throws an error. We are mixing two different colour scales, one for the points and one for the line.

We can try to fix it declaring a new scale `new_scale_color`

```
plot_observed_data + new_scale_color() +
  geom_ribbon(data=pred.data, aes(x=log.area, ymin = pred.lower, ymax = pred.upper, fill=rain.sign), al
      geom_line(data=pred.data,aes(x=log.area,y=prediction,colour=rain.sign) ) + coord_cartesian(ylim =
```

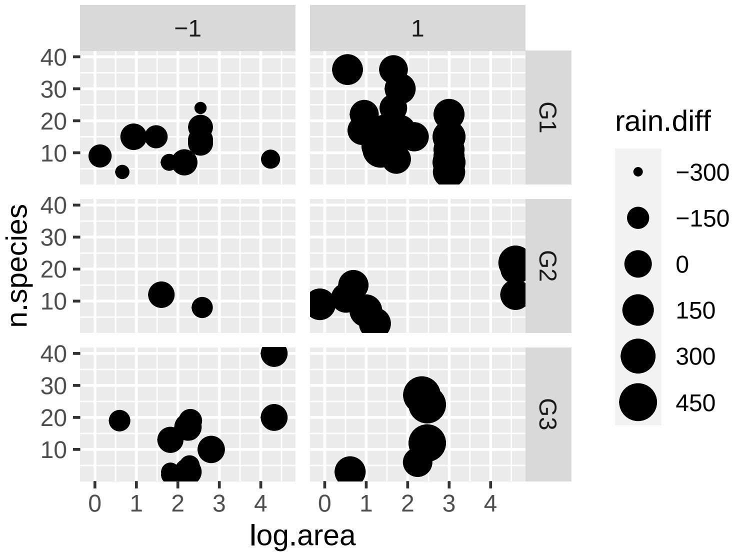


That looks… confusing!?. The plot is not that bad, but the legends are a mess. We get again separate size and colour legends for the points, and separate legends for lines and ribbon. You can spend hours trying to fix this, or maybe we should simplify again the plot and reduce the aesthetics that are redundant.

Let's get back to the beginning and simplify the observed point data first. Shapes are redundant with the facets, and colour and size are also redundant:

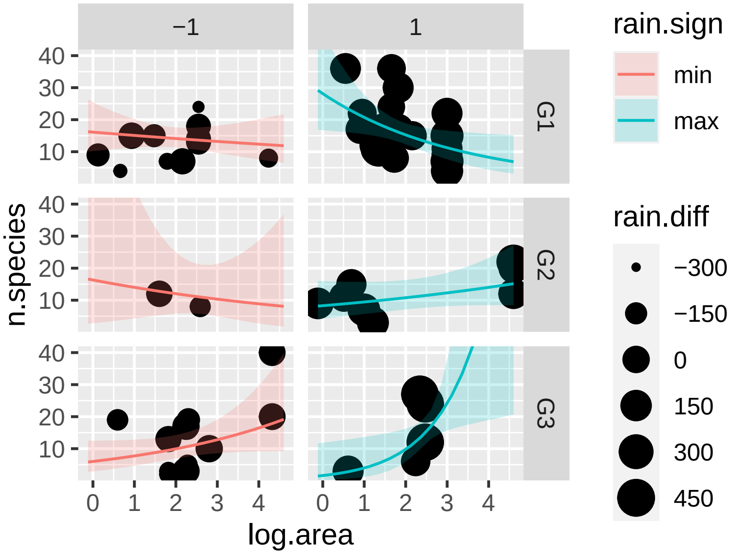
```
plot_with4vars <- ggplot(data=obs.data) +
  geom_point(aes(x=log.area,y=n.species, size=rain.diff)) + facet_grid(blocks~rain.sign)

(plot_observed_data <- plot_with4vars  +
scale_size_continuous(limits=var3.limits, breaks=var3.breaks) )
```



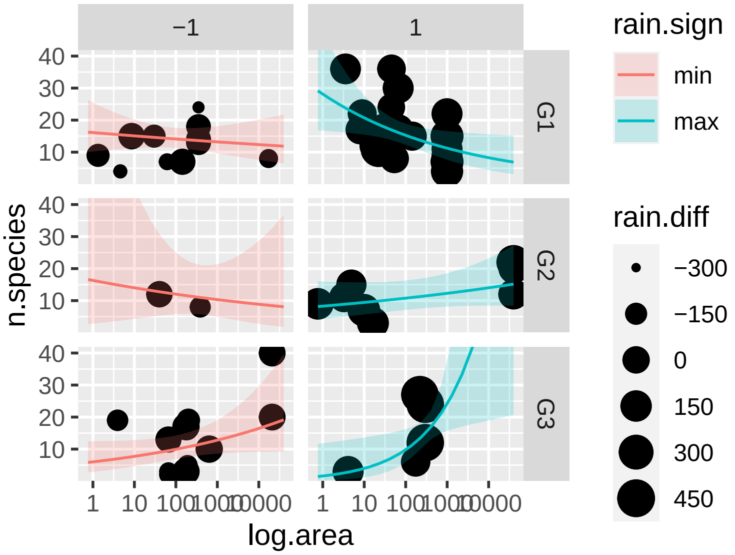
We can now add the lines and ribbons without conflict with previous colour scales

```
(plot_obs_and_pred_data <-
  plot_observed_data +
    geom_ribbon(data=pred.data, aes(x=log.area, ymin = pred.lower, ymax = pred.upper, fill=rain.sign),
     geom_line(data=pred.data,aes(x=log.area,y=prediction,colour=rain.sign) ) + coord_cartesian(ylim =
```



This looks good, but we suddently realize the scale in the x axes is a logarithm of area, but we want to show the real value of area (in hectars). Should we go back and fix the data and model and run all the code again?… or maybe we just need to add one line of code to fix the plot:

```
plot_obs_and_pred_data + scale_x_continuous( breaks=seq(0,4), labels=c(1,10,100,1000,10000))
```



Important: this fixes the looks in the plot, but the model is still using the log.area as a variable, this is ok for this example but might not be so in other contexts. Make sure you add accurate descriptions in methods and figure legends when you work in your research project.

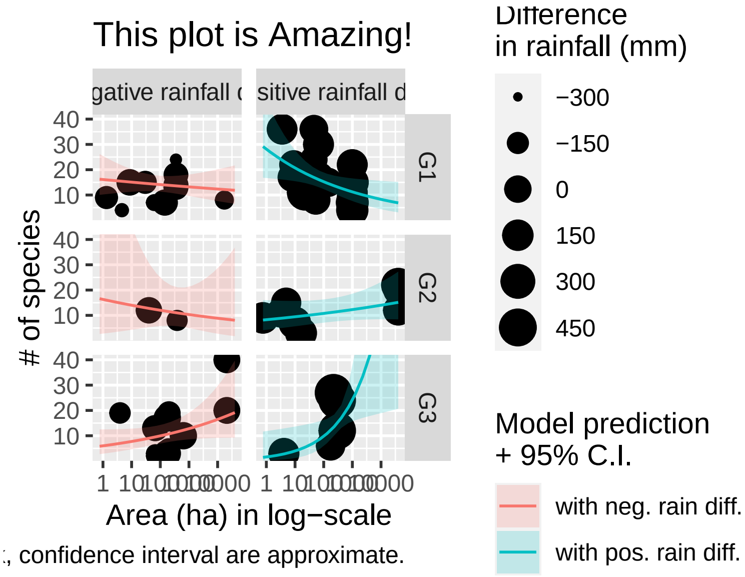
Now the plot looks good enough, all the information we want is in there, and we discarded some redundant features.

As a final step we can fix the legends and labels to make them more informative, notice the use of several functions to add or overwrite labels:

```
plot_obs_and_pred_data +
  facet_grid(blocks~rain.sign, labeller = labeller(rain.sign = c(`-1`='Negative rainfall diff.',`1`='Po
  scale_x_continuous( breaks=seq(0,4), labels=c(1,10,100,1000,10000))  +
  labs(title='This plot is Amazing!',caption='Prediction from a GLM with Quasipoisson error distributio
  labs(colour='Model prediction \n+ 95% C.I.', fill='Model prediction \n+ 95% C.I.',size='Difference\nir
  scale_colour_discrete(labels=c("with neg. rain diff.","with pos. rain diff.")) +
  scale_fill_discrete(labels=c("with neg. rain diff.","with pos. rain diff."))
```

```
## Scale for 'colour' is already present. Adding another scale for 'colour',
## which will replace the existing scale.
```

```
## Scale for 'fill' is already present. Adding another scale for 'fill', which
## will replace the existing scale.
```



Useful links

- https://fromthebottomoftheheap.net/2017/05/01/glm-prediction-intervals-i/
- https://www.datanovia.com/en/blog/how-to-change-ggplot-facet-labels/

Debugging info

Check the R session info:

```
sessionInfo()
```

```
## R version 4.1.0 (2021-05-18)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS Big Sur 10.16
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.1/Resources/lib/libRblas.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.1/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_AU.UTF-8/en_AU.UTF-8/en_AU.UTF-8/C/en_AU.UTF-8/en_AU.UTF-8
##
## attached base packages:
```

```
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] ggnewscale_0.4.5 ggplot2_3.3.5
##
## loaded via a namespace (and not attached):
##  [1] highr_0.9         pillar_1.6.2      compiler_4.1.0    tools_4.1.0
##  [5] digest_0.6.27     evaluate_0.14     lifecycle_1.0.0   tibble_3.1.3
##  [9] gtable_0.3.0      pkgconfig_2.0.3   rlang_0.4.11      DBI_1.1.1
## [13] yaml_2.2.1        xfun_0.24         withr_2.4.2       stringr_1.4.0
## [17] dplyr_1.0.7       knitr_1.33        generics_0.1.0    vctrs_0.3.8
## [21] grid_4.1.0        tidyselect_1.1.1  glue_1.4.2        R6_2.5.1
## [25] fansi_0.5.0       rmarkdown_2.9     purrr_0.3.4       farver_2.1.0
## [29] magrittr_2.0.1    scales_1.1.1      ellipsis_0.3.2    htmltools_0.5.1.1
## [33] assertthat_0.2.1  colorspace_2.0-2  labeling_0.4.2    utf8_1.2.2
## [37] stringi_1.7.3     munsell_0.5.0     crayon_1.4.1
```