Estimate Robustness of Real Networks

Andrew L Jackson, Jean-Francois Arnoldi, Sam P. Ross & Ian Donohue

09 April 2020

```r
library(tidyverse)
```

```
## -- Attaching packages ------------------------------------------------------------------- tidyverse 

## v ggplot2 3.3.0     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.5
## v tidyr   1.0.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0

## -- Conflicts ---------------------------------------------------------------------- tidyverse_confli
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(furrr) # for parallel implmenetation of purrr
```

```
## Loading required package: future
```

```r
library(latex2exp) # for latex style equations in figures
```

Import the Web of Life data

Loop over the datasets and import them into a list.

```r
# get all file names in the folder data/
all_files <- dir("data/")

# find the indices that are README or reference files for each web type
remove_these <- c(grep("README", all_files), grep("references", all_files))

# remove them from the vector of names and prepend the data/ folder address
all_web_files <- paste0("data/", all_files[-remove_these])

# import all the webs and covert then to binary association form
all_webs <- map(all_web_files, ~sign(read.csv(.x)))

# add names to each of the list entries. Required by map()
names(all_webs) <- all_web_files

# a table of all the webs by type - used to create summary counts of each
# web type.
types <- unlist(map(names(all_webs), ~substr(.x, 8, 9)))
n_types <- table(types)
```

We used all the available bipartite webs from http://www.web-of-life.es as downloaded on 18 Febrary 2020. The original matrices included measures of times recorded, which we converted to binary association matrices.

Each interaction matrix types described below comprise frist the species in rows (S) and second the traits (N) by columns for example for species of Anemone (rows) provide habitat for Fish (columns):

- AF = Anemone - Fish (n = 17)
- HP = Host - Parasite (n = 51)
- PA = Plant - Ant (n = 4)
- PH = Plant - Herbivore (n = 4)
- PL = Plant - Pollinator (n = 148)
- SD = Seed - dispersers (n = 34)

Summary statistics

Define some functions that we will use to generate summary statics

A function to calculate our measure of fragility f,

$$f = \frac{1}{S}\frac{\log(N)}{|\log(q)|}$$

where q=1-p and p is the connectance estimated as p~#links/(SxN).

```
fragilityEstimate <- function(A) {

  # rows are traits
  N <- nrow(A)

  # columns are species
  S <- ncol(A)

  # connectance measures
  p <- sum(A) / (S * N)
  q <-  1 - p

  # fragility
  # lines below are equivalent
  # f <- (1/S) * (log(N) / abs(log(q))) # alternative parameterisation
  f <- -log(N) / log(q) / S


  return(f)

}
```

A function to calculate the robustness of a given web, defined by its association matrix **A** (whose elements $a_{ij} = \{0, 1\}$) by simulating extinctions of species until at least one trait is lost, which owing to our assumption of $E^* =$ AND also equates to loss of the higher level service.

```
robustness <- function(Aperm, S = ncol(Aperm)){

  nn <- 0
  mm <- 0

  # move along columns from left to right until we lose a trait
  while (mm==0 & nn <= S) {
    nn <- nn + 1
    mm <- min(rowSums( as.matrix(Aperm[, 1:nn])))
  }

  return( (S-nn+1) / (S * 1.0))
}
```

The function `shuffleExtinctionSingle()` takes a given empirical web, permute its columns, and calculate its robustness using the function `robustness` defined above. The function `sampleRobustness()` loops the call to `shuffleExtinctionSingle()` using parallel computing via the function `furr::future_map_dbl()` and returns the mean robustness across the defined number of replicates (we use `nb = 500` as defined later).

```
shuffleExinctionSingle <- function(A, S) {

  # generate a permuation for species in A
  species <- sample(S)

  # premute within rows
  Aperm <- A[,species]

  # calculate robustness
  rob <- robustness(Aperm)

  return(rob)

}

sampleRobustness <- function(A, S = ncol(A), nb = 10){

  # loop over replicates using parallel furrr map function.
  res <- future_map_dbl(1:nb, ~shuffleExinctionSingle( A, S))

  # return the mean of the results vector which is mean of
  return(mean(res))

}
```

A gaussian-exponential function to estimate robustness from fragility $R = \exp(-f - \frac{2}{3}x^2)$

```
predictRobustness <- function(x) {exp(-x - (2/3)* x ^2 )}
```

Apply our functions to each web

We calculate summary statistics on each web and convert to data.frame format. Beyond the basic statistics covered in our paper we are interested in quantifying how the empirical web depart from the random expectation of links as assumed in our basic theory. If we assume that species S are connected to traits N according to a binomial distribution then we calculate how the empirical networks deviate from this prediction. Speficically, the first two moments of the binomial distribution (with random variable x) are given by $\bar{x} = pq$ and $\mathrm{Var}(x) = npq$ where $q = 1 - p$. We can then define Dispersion d as the proportional deviance from 1 of the sample estimated via

$$d = \frac{1}{q}\frac{\mathrm{Var}(S_n)}{\bar{S_n}}$$

```
# map over all the webs and calculate some of the fundamental statistics
# used in our theory. Retuns a data.frame via map_dfr().
# Add a column designating the web type at the end from the vector
# types we created early on when reading the file names.
df_webs <-  all_webs %>%
  map_dfr(~data.frame(S = as.numeric(ncol(.x)),
                      N = as.numeric(nrow(.x)),
                      sum_A = as.numeric(sum(.x)),
                      min_S_per_N = min(rowSums(.x)),
                      var_S_per_N = var(rowSums(.x)),
                      mean_S_per_N = mean(rowSums(.x)),
                      est_p = sum(.x) / (ncol(.x) * nrow(.x)),
                      fragility = fragilityEstimate(.x)),
                      .id = "file") %>%
  mutate(web_type = types)

# calculate dispersion, where the expected value according to the binomial
# distribution is 1.
df_webs <- df_webs %>%
  mutate(dispersion = (var_S_per_N / mean_S_per_N) *
                        (1 / 1 - est_p))
```

Simulate Extinctions to calculate robustness

We now simulate extinctions on the randomly permuted empirical webs using our functions defined above. This is computationally intensive, even with parallel computing implemented on the loop, and so we introduce a toggle variable `do_robustness` which can either load a previous run from file `do_robustness == FALSE`, or otherwise invoke the simulations.

```
# number of replicated
n_samples <- 500

# a logical toggle to determine whether the computationally expensive
# extinctions should be run. They can be loaded from file if they already
# exist.
do_robustness <- FALSE

if (do_robustness == TRUE ) {

  # set up the multicore for the function sampleRobustness which calls them.
  # im not sure if i should do this within sampleRobustness or if its ok to
  # do it once here outside.
  plan(multisession(workers = 3))

  # map over all the webs and sample robustness
  robustness_results <- all_webs %>%
    map_dbl(~sampleRobustness(.x, nb = n_samples))

  # save to the results to file
  save(robustness_results, file = "robustness_run.rda", compress = "xz")


}


# Load from file if toggled
if (do_robustness == FALSE){
```

```r
  print("NB sampling for robustness not run. Loaded instead from previous run.")
  load("robustness_run.rda")
}
```

```
## [1] "NB sampling for robustness not run. Loaded instead from previous run."
```

```r
# Add the robustness estimate to the data.frame and
# calcuate the residual to Jeff's model
df_webs <- df_webs %>%
  mutate(robustness = robustness_results)

# add the estimated fragility for each network based on their fragility
df_webs <- df_webs %>% mutate(robustness_hat = predictRobustness(fragility))
```

Random sampling of extinctions until loss of any trait was simulate with $n = 500$ on each of the empirical interaction networks and the mean proportion of extinctions taken as the robustness R for that network.

Filter webs

We remove webs that have only 1 trait (n = 3) and webs that show no variation in the number of species S per trait N (n = 4, and all of which are Anemone-Fish type webs, all of which had anemones being associated with only 1 or 0 species of fish). Filtering is done after the simulations in order to preserve the alignment of the web file name and the result. [*AJ 2020-4-9: I have subsequently added the file name to the df_webs object and so we could filter earlier in the process*]

```r
# remove webs that dont meet our criteria
df_webs <- df_webs %>% filter( dispersion != 0.0 & !is.na(var_S_per_N) )
```

Plot results

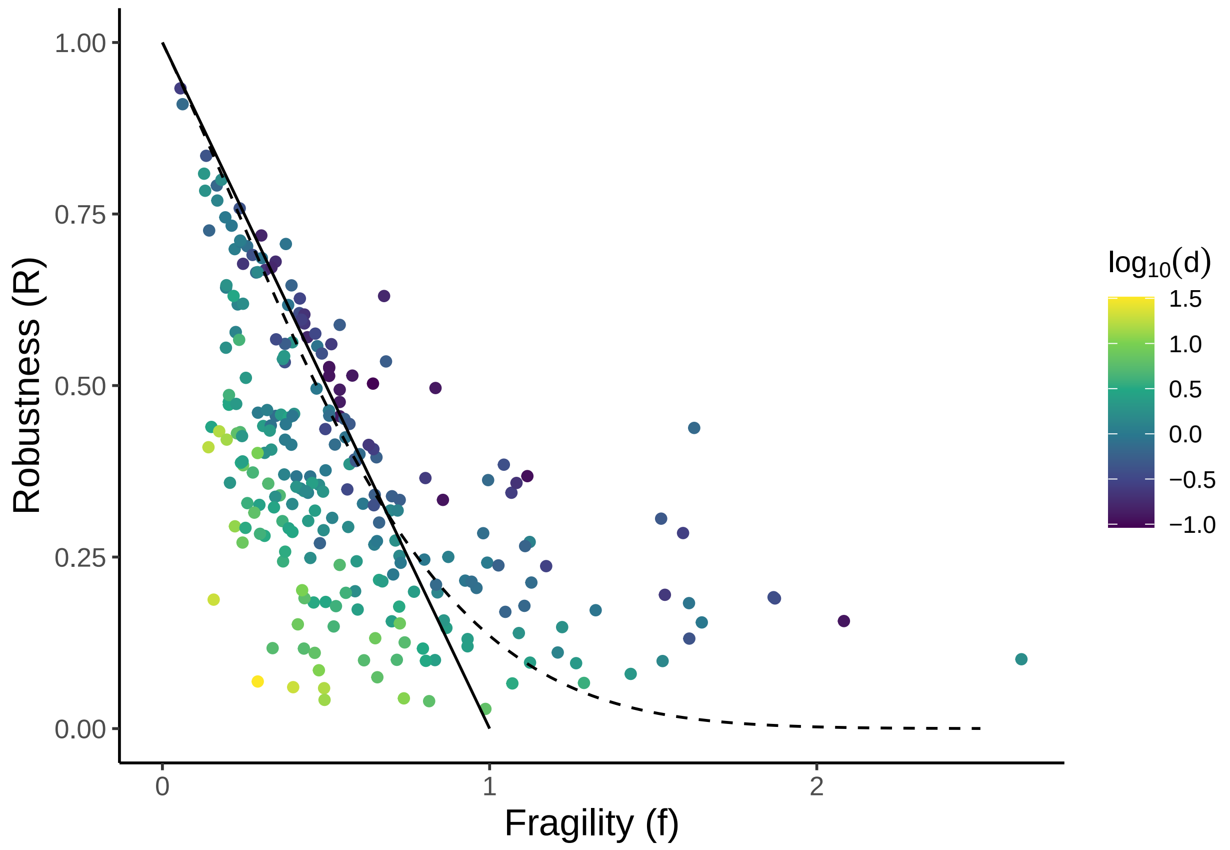
Here we set some plotting parameters for consistency across panels.

```r
axis_text_size <- 10
axis_title_size <- 14
```

In order to plot the analytical model, we create a dataframe object of a sequence of fragility values spanning out observed range and the corresponding predicted robustness ($\hat{R}$).

```r
new_f <- seq(from = 0, to = 2.5, length.out = 100)

# the estimated Robusteness (Rhat)
est_r <- exp(-new_f - new_f^2) #predictRobustness(new_f)

fit_df <- data.frame(new_f = new_f,
                     est_r = est_r)
```

```r
g1 <- ggplot(df_webs, aes(x = fragility,
                     y = robustness,
                     color = log10(dispersion))) +
  geom_point() +
    scale_color_viridis_c() +
  geom_path(data = data.frame(xx = c(0, 1),
                              yy = c(1, 0)),
           mapping = aes(x = xx, y = yy),
           color = "black") +
  geom_line(data = fit_df, mapping = aes(x = new_f, y = est_r, color = NULL),
           linetype = 2) +
  ylab("Robustness (R)") +
  xlab("Fragility (f)") +
   labs(color = expression(log[10](d))) +
  theme_classic() +
  theme(axis.text = element_text(size = axis_text_size),
        axis.title = element_text(size = axis_title_size))


print(g1)
```



```
#
ggsave(g1,filename = "images/emp-robust-fragility.png",
       width = 20, height = 12, units = "cm",
       scale = 1)
```

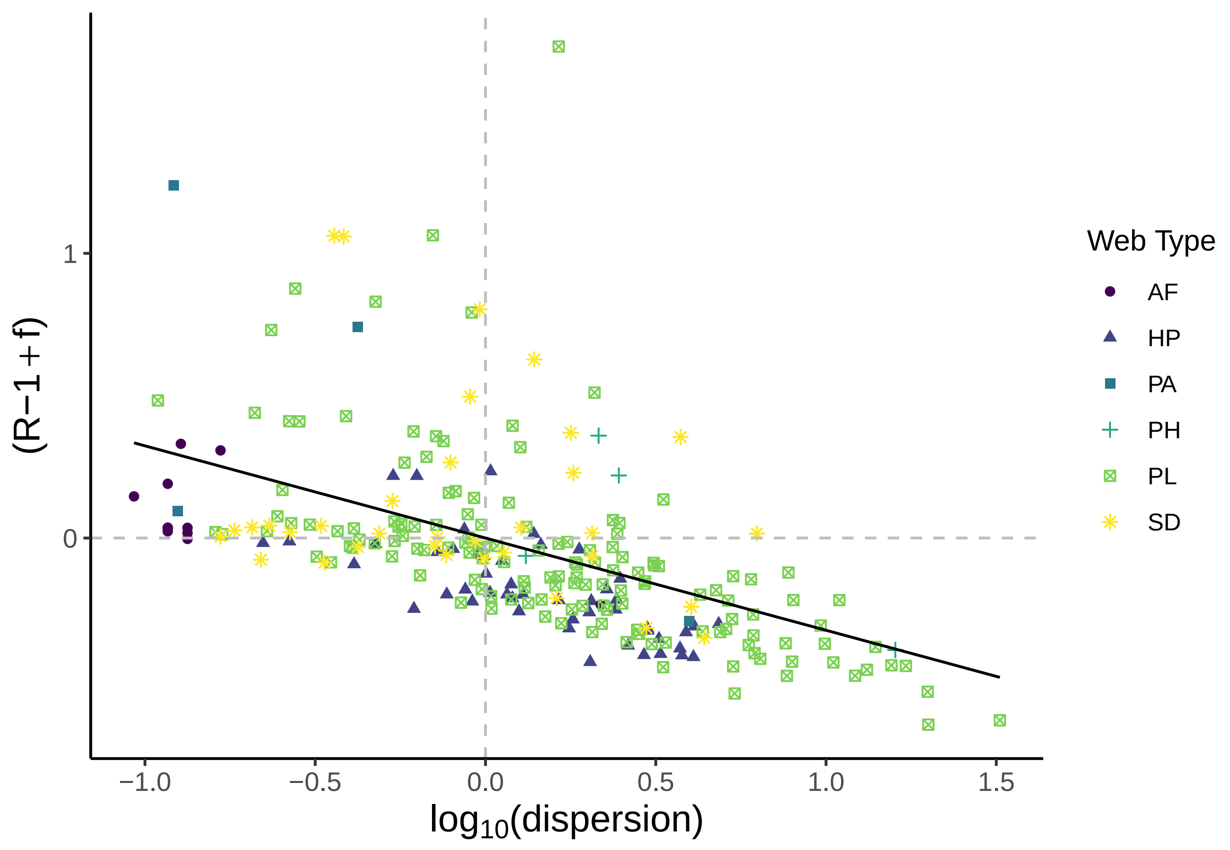
Figure S2.1. The relationship between simulated robustness and analytically esimated fragility as calculated on the $n = 251$ empirical networks matches close the analytical approximation (black line). Each network is coloured by the degree of disperion (d) (log10 scale) according to a binomial distribution, where values below 0 represent under-dispersion, 0 is ideally dispersed and values greater than 0 indciate over-dispersion. The solid black line indicates the approximately linear relationship of $R = 1 - f$ for $f < 1$, and the dotted black line the quadratic relationship $R = \exp(-f - f^2)$

Fit the linear model and calculate the residuals to this fit.

```
# add columns for the residuals to the expected R = 1 - fragility
# relationship; log10(dispersion); and a factor type of web_type variable.
df_webs <- df_webs %>% mutate(residRf = robustness -1 + fragility,
                              log10d = log10(dispersion),
                              web_type_fac = as.factor(web_type))


# model of the residuals of R = 1-f tor f<1 on log(dispersion)
mRf <- lm( residRf ~ -1 + log10d,
           data = df_webs %>% filter(fragility < 1))

# when fitting without an intercept we need to manually add in a = 0
cc <- c(0, coef(mRf))


# Use broom::augment() to predict the residuals for the entire dataset, i.e.
# for all data including f<1 which was used to fit, but also f>1. augment()
# adds it on to the existing data and by default retains all variables.
df_webs <- broom::augment(mRf, newdata = df_webs)
```

Plot the residuals of $R = 1 - f$ against $\log_{10}(dispersion)$.

```
g3 <- ggplot(df_webs,
                        aes(x = log10(dispersion),
                            y = residRf,
                            color = web_type_fac,
                            shape = web_type_fac)) +
  geom_point() +
  scale_color_viridis_d() +
  geom_hline(yintercept = 0, color = "grey", linetype = 2) +
  geom_vline(xintercept = 0, color = "grey", linetype = 2) +
  geom_line(mapping = aes(x = log10d,
                          y = .fitted,
                          color = NULL,
                          shape = NULL)) +
  labs(color = "Web Type",
       shape = "Web Type") +
  ylab(TeX("$(R -1 + f)$")) +
  xlab(TeX("$\\log_{10}(\\dispersion)$")) +
  theme_classic() +
  theme(axis.text = element_text(size = axis_text_size),
        axis.title = element_text(size = axis_title_size)) +
guides(color = guide_legend(override.aes = list(linetype = 0)))


print(g3)
```



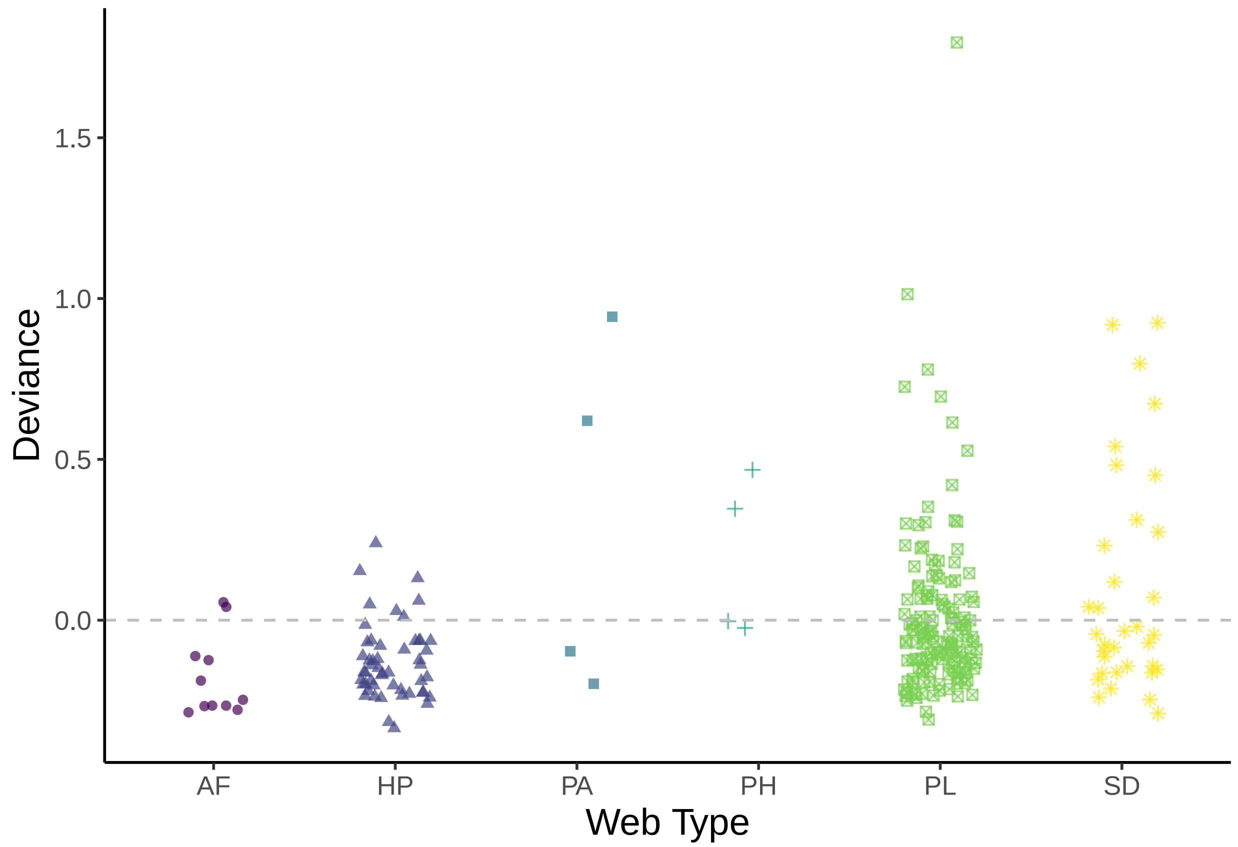
```r
ggsave(g3,
       filename = "images/emp-resid-dispersion.png",
       width = 20, height = 12, units = "cm")
```

Plot these two figures together using `patchwork`

```r
# g_ab <- (g1 | g3_simplified)
#
# print(g_ab)
#
# ggsave(filename = "frag-robust-disp.png", plot = g_ab,
       # width = 15, height = 9, units = "cm")
```

Figure S2.2. The relationship between the residuals to the relationship $R = 1 - f$ and $\log_{10}(d)$ is modelled as a linear relationship through the origin. The The slope was estimated by fitting to all points in the range $f < 1$ where the relationship is approximately linear (Figure S2.1). Different web types are indicated by colour and shape. The origin is shown for clarity and reference as dotted grey lines.

```r
g4 <- ggplot(df_webs,
             aes(x = web_type_fac,
                 y = residRf - .fitted)) +
  # geom_boxplot(outlier.shape = NA) +
  geom_jitter(mapping = aes(color = web_type_fac,
                            shape = web_type_fac),
              width = 0.2,
              alpha = 0.7) +
  scale_color_viridis_d() +
  geom_hline(yintercept = 0, color = "grey", linetype = 2) +
  theme_classic() +
  theme(axis.text = element_text(size = axis_text_size),
        axis.title = element_text(size = axis_title_size)) +
  guides(color = FALSE, shape = FALSE, alpha = FALSE) +
  xlab("Web Type") +
  ylab("Deviance")

print(g4)
```



```
ggsave(g4,filename = "images/emp-resid-dispersion-by-web.png",
       width = 20, height = 12, units = "cm",)
```

Figure S2.3. The deviance from the expected linear relationship of $(R - 1 + f) = b\log_{10}(d)$ is a reasonable universal approximation across the different web types (colours and shapes match those in Figure S2.2 for clarity).

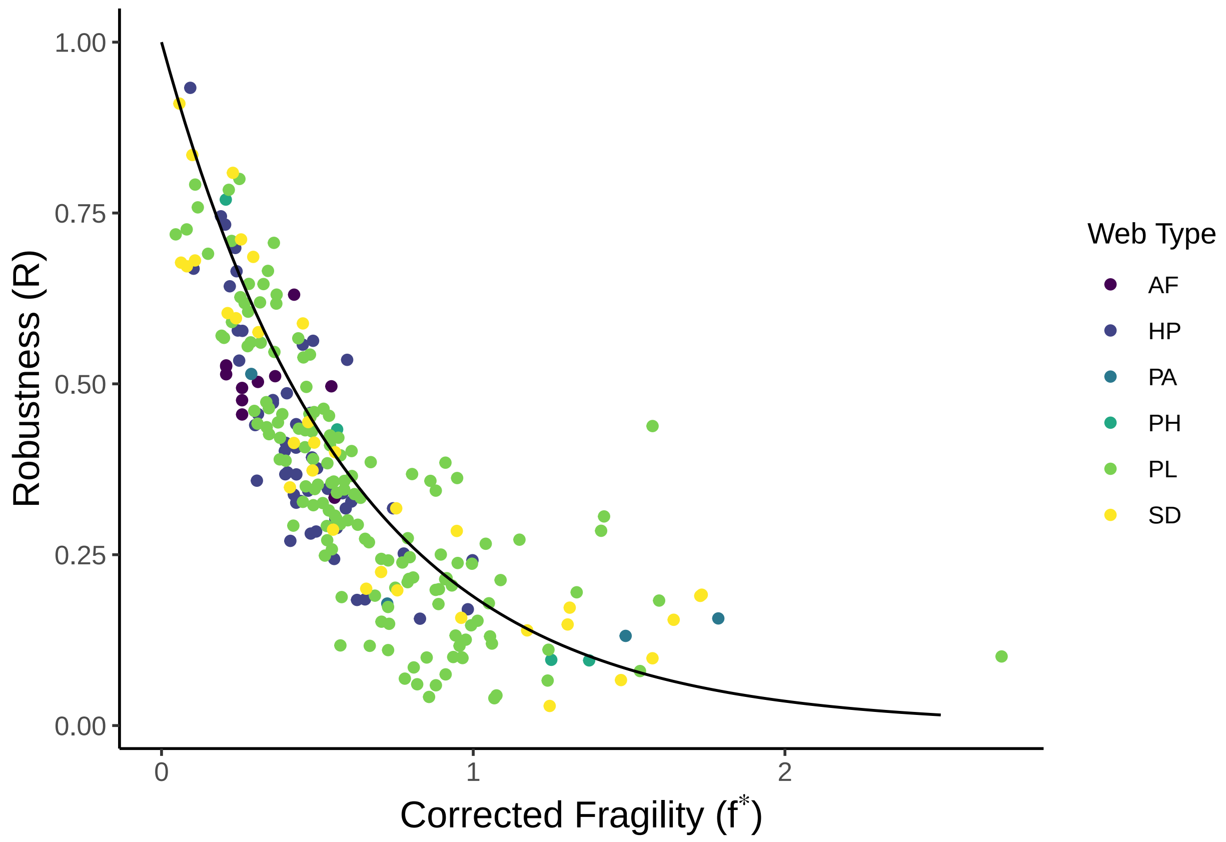
Calculate modified measure of fragility f^*

We can then use a linear model as fit to Figure S2.1b to account for dispersion for a new metric that describes Robustness (R) given Fragility (f) and Dispersion (d) of a given network.

```
# regress robsustness against (1 - fragility) and then esimate the coefficient
# of the linear regression of the residuals to this plot against
# log10(dispersion) and calculate
# fstar = fragility +  * log10(dispersion)
df_webs <- df_webs %>%
  mutate(fstar = pmax(fragility - .fitted, runif(n(), 0, 0.1)))
```

Generate a plot of $R \propto f^*$

```
g6 <- ggplot(df_webs, aes(x = fstar,
                          y = robustness,
                          color = web_type_fac)) +
  geom_point() +
  scale_color_viridis_d() +
  geom_line(data = fit_df, mapping = aes(x = new_f, y = exp(-5/3*new_f), color = NULL)) +
  ylab("Robustness (R)") +
  xlab(TeX("Corrected Fragility $(f^*)$")) +
  labs(color = "Web Type", line.style = NULL) +
  theme_classic() +
  theme(axis.text = element_text(size = axis_text_size),
        axis.title = element_text(size = axis_title_size)) +
  guides(color = guide_legend(override.aes = list(linetype = 0)))

print(g6)
```



```
ggsave(g6, filename = "images/emp-R-by-fstar.png",
       width = 20, height = 12, units = "cm")
```

Figure S2.3. The relationship betwenn Robustness (R) and corrected Fragility measure approximated by $f^* = f +$
$log_{10}(d)$, where $= 0.32$. The

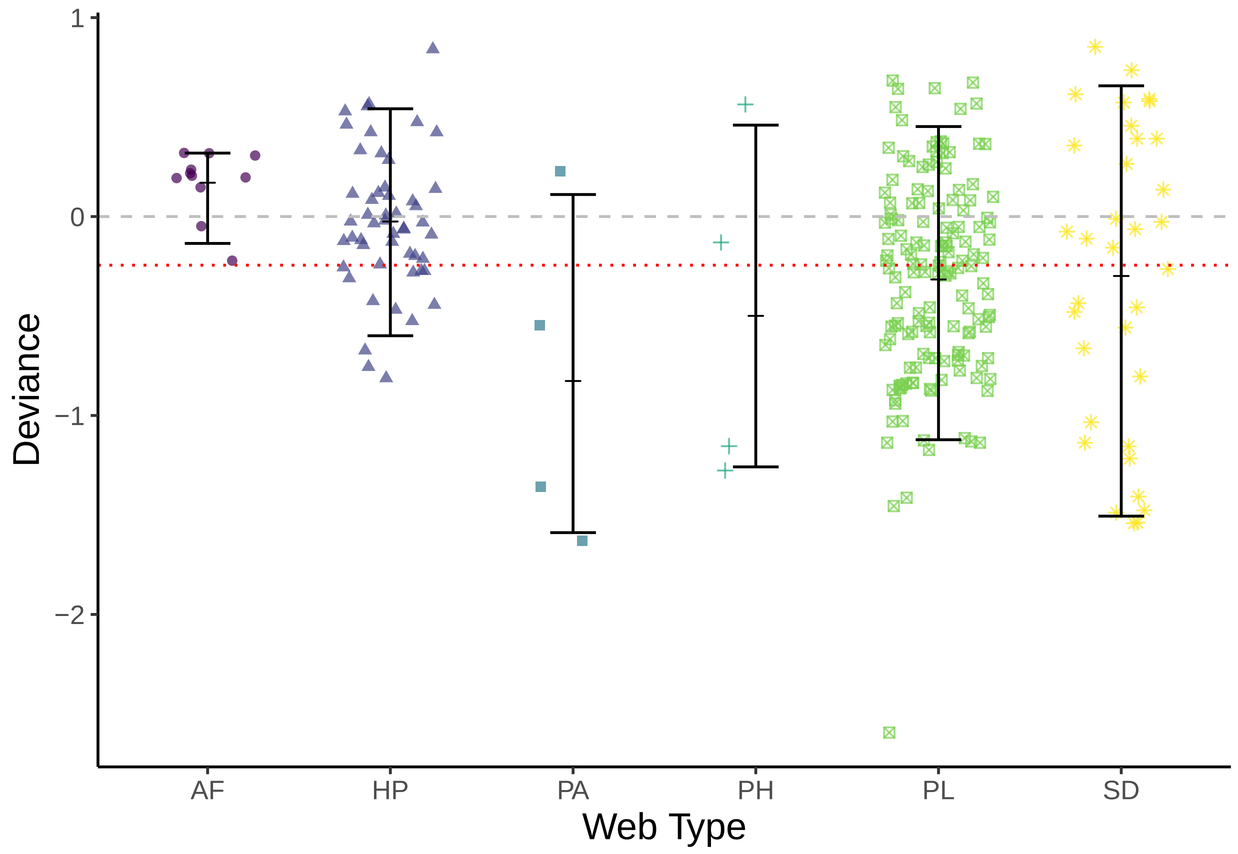
The correlation coefficients for Robustness (R) with the various estimates are:

- cor(R, f) = -0.64
- cor$(R, log(d))$ = -0.47
- cor(R, f^*) = -0.88

Residual variance by web type

We explore the residual variance from our define relationship between R and f^* by web type, in order to assess whether there is additional structure in these webs that might further explain departures from the predicted relationship. Across all web types there is an overall trend towards our relationship over-esimating robustness slightly (red dotted line), however the predicted value falls within the 90% prediction interval for all web types (errorbars), although three webs have relatively low replicates: PA and PH have $n = 4$; and AF has $n = 11$. With greater replicates, it may be possible to confirm and explore further any variation among web type; for example both Anenome-Fish (AF) and Host-Parasite (HP) webs show some tendency towards hiher robustness for a given fragility (f^*) and dispersion (d) compared with Plant-Pollinator (PL) and Seed-Dispersers (SD).

```r
df_webs <- df_webs %>% mutate(resid_fstar = robustness - fstar)

summary_by_web_type <- df_webs %>%
  group_by(web_type_fac) %>%
  summarise(mu = mean(resid_fstar),
            sd = sd(resid_fstar),
            se = sd(resid_fstar) / sqrt(n()),
            qa = quantile(resid_fstar, probs = 0.05),
            qb = quantile(resid_fstar, probs = 0.95))



g9 <- ggplot(df_webs,
             aes(x = web_type_fac,
                 y = resid_fstar)) +
  # geom_boxplot(outlier.shape = NA) +
  geom_jitter(mapping = aes(color = web_type_fac,
                            shape = web_type_fac),
              width = 0.3,
              alpha = 0.7) +
  scale_color_viridis_d() +
  geom_hline(yintercept = 0, color = "grey", linetype = 2) +
  theme_classic() +
  theme(axis.text = element_text(size = axis_text_size),
        axis.title = element_text(size = axis_title_size)) +
  guides(color = FALSE, shape = FALSE, alpha = FALSE) +
  xlab("Web Type") +
  ylab("Deviance") +
  geom_errorbar(data = summary_by_web_type,
                mapping = aes(x = web_type_fac,
                              y = NULL,
                              ymin = qa,
                              ymax = qb ,
                              color = NULL),
                color = "black",
                width = 0.25) +
  geom_point(summary_by_web_type,
             mapping = aes(x = web_type_fac,
                        y = mu,
                        color = NULL),
             shape = 3) +
  geom_hline(yintercept = mean(df_webs$resid_fstar),
             color = "red",
             linetype = 3)

print(g9)
```



```
ggsave(g9,filename = "images/emp-resid-fstar-by-web.png",
       width = 20, height = 12, units = "cm",)
```

Save results to file

write the dataframe to file for sharing

```
# and write to file
write_csv(df_webs, path = "web_statistics.csv")
```