Selection on within-individual variation in flowering time in Lathyrus vernus

Data preparation

Alicia Valdés

31 January, 2023

Contents

Session info 16

Read data for individuals from Excel file

```
data_ids_87 <- read_excel("data/edited/individual_characteristics.xlsx",
                          sheet = "1987_editsAV")
data_ids_88 <- read_excel("data/edited/individual_characteristics.xlsx",
                          sheet = "1988_editsAV")
data_ids_89 <- read_excel("data/edited/individual_characteristics.xlsx",
                          sheet = "1989_editsAV")
```

Read data for individual flowers from Excel files

```
data_id_flowers_87 <- read_excel("data/edited/individual_flower_characteristics.xlsx",
                          sheet = "1987")
data_id_flowers_88 <- read_excel("data/edited/individual_flower_characteristics.xlsx",
                          sheet = "1988")
data_id_flowers_89 <- read_excel("data/edited/individual_flower_characteristics.xlsx",
                          sheet = "1989")
data_Ind2_33_1989 <- read_excel("data/edited/individual_characteristics_Ind2_33_1989.xlsx",
                                   sheet = "to_R")
```

```
nrow(data_ids_87)
```

```
## [1] 231
```

```
# 231 rows
nrow(data_ids_88)
```

```
## [1] 169
```

```
# 169 rows
nrow(data_ids_89)
```

```
## [1] 96
```

```
# 96 rows
```

Error in 1989

Ind = 2:33 appears twice, there was a problem with this Ind, I will remove those two records and add a new record for Ind 2:33 with all moments that were recalculated in Excel.

```r
subset(data_ids_89,Ind=="2:33")
```

```
## # A tibble: 2 x 17
##      ID Subplot Ind   `Mean (MFD)`    SD   Skew Kurtosis `Max (LFD)` `Min (FFD)`
##   <dbl>   <dbl> <chr>        <dbl> <dbl>  <dbl>    <dbl>       <dbl>       <dbl>
## 1    42       2 2:33          1.60  1.08  0.900   -0.170        3.84       0.518
## 2    52       2 2:33          3.84 NA    NA       NA            3.84       3.84
##   `Range (Duration)` `Flower N` Fruits `Fruit init (fr/fl)` `Total seeds`
##                <dbl>      <dbl>  <dbl>                <dbl>         <dbl>
## 1               3.32         17      8                0.471            27
## 2               0             1      0                0                 0
##   `Preyed seeds` `Intact seeds (fitness)` Imputed
##            <dbl>                    <dbl>   <dbl>
## 1           6.36                     20.6       0
## 2           0                         0         0
```

```r
data_ids_89 <- data_ids_89 %>% filter(!(Ind=="2:33"))
data_ids_89 <- bind_rows(data_ids_89,data_Ind2_33_1989)
```

Rename columns

```r
data_ids_87 <- data_ids_87 %>%
  rename(number = ID, subplot = Subplot, id = Ind, avFD = `Mean (MFD)`,
         skew =  Skew, kurt = Kurtosis,LFD = `Max (LFD)`, FFD = `Min (FFD)`,
         dur = `Range (Duration)`, n_fl = `Flower N`, n_fr = Fruits,
         fr_init = `Fruit init (fr/fl)`, n_seed = `Total seeds`,
         n_preyed_seed = `Preyed seeds`,
         fitness = `Intact seeds (fitness)`,
         imp_seed_preyed = Imputed,
         n_seed_per_fr = `Seeds per fruit`,
         prop_seed_preyed = `Proportion preyed`)
data_ids_88 <- data_ids_88 %>%
  rename(number = ID, subplot = Subplot, id = Ind, avFD = `Mean (MFD)`,
         skew =  Skew, kurt = Kurtosis,LFD = `Max (LFD)`, FFD = `Min (FFD)`,
         dur = `Range (Duration)`, n_fl = `Flower N`, n_fr = Fruits,
         fr_init = `Fruit init (fr/fl)`, n_seed = `Total seeds`,
         n_preyed_seed = `Preyed seeds`,
         fitness = `Intact seeds (fitness)`,
         imp_seed_preyed = Imputed,
         n_seed_per_fr = `Seeds per fruit`,
         prop_seed_preyed = `Proportion preyed`)
data_ids_89 <- data_ids_89 %>%
  rename(number = ID, subplot = Subplot, id = Ind, avFD = `Mean (MFD)`,
         skew =  Skew, kurt = Kurtosis,LFD = `Max (LFD)`, FFD = `Min (FFD)`,
         dur = `Range (Duration)`, n_fl = `Flower N`, n_fr = Fruits,
         fr_init = `Fruit init (fr/fl)`, n_seed = `Total seeds`,
         n_preyed_seed = `Preyed seeds`,
         fitness = `Intact seeds (fitness)`,
         imp_seed_preyed = Imputed)
```

Calculate number of seeds per fruit and proportion of seeds preyed in 1989

```
data_ids_89 <- data_ids_89 %>%
  mutate(n_seed_per_fr = ifelse(n_fr==0, NA, n_seed / n_fr),
         prop_seed_preyed = ifelse(n_seed==0, NA, n_preyed_seed / n_seed))
```

Change column types

```
data_ids_87 <- data_ids_87 %>%
  mutate(imp_seed_preyed = as.factor(imp_seed_preyed))
data_ids_88 <- data_ids_88 %>%
  mutate(imp_seed_preyed = as.factor(imp_seed_preyed))
data_ids_89 <- data_ids_89 %>%
  mutate(imp_seed_preyed = as.factor(imp_seed_preyed))
# See if I keep integer values as "double"!
```

Recalculate moments with individual flower data

Data prep individual flower data

```
data_id_flowers_87 <- data_id_flowers_87 %>%
  select(RUTA,GENET...2,`New Phenoadj based on intervals`) %>%
  mutate(id = paste(RUTA,GENET...2,sep=":"),
         opening_date = `New Phenoadj based on intervals`) %>%
  rename(subplot = RUTA, number = GENET...2) %>%
  select(-`New Phenoadj based on intervals`)
data_id_flowers_88 <- data_id_flowers_88 %>%
  select(RUTA...1,GENET...2,`New Phenoadj based on intervals`) %>%
  mutate(id = paste(RUTA...1,GENET...2,sep=":"),
         opening_date = `New Phenoadj based on intervals`) %>%
  rename(subplot = RUTA...1, number = GENET...2) %>%
  select(-`New Phenoadj based on intervals`) %>%
  filter(!(subplot==8|subplot==9))
data_id_flowers_89 <- data_id_flowers_89 %>%
  select(RUTA,GENET,`Corrected pheno`) %>%
  mutate(id = paste(RUTA,GENET,sep=":"),
         opening_date = `Corrected pheno`) %>%
  rename(subplot = RUTA, number = GENET) %>%
  select(-`Corrected pheno`)
```

See if the number of individuals in each subplot matches betwen individual data and individual flower data.

```
data_ids_87%>%group_by(subplot)%>%summarise(n_indiv=n())
```

```
## # A tibble: 6 x 2
##   subplot n_indiv
##     <dbl>   <int>
## 1       1      76
## 2       2      25
## 3       3      60
## 4       4      23
## 5       5      28
## 6       6      19
```

```
data_id_flowers_87%>%group_by(subplot)%>%summarise(n_indiv=n_distinct(id))
```

```
## # A tibble: 6 x 2
##   subplot n_indiv
##     <dbl>   <int>
## 1       1      76
## 2       2      25
## 3       3      60
## 4       4      23
## 5       5      28
## 6       6      19
```

```
data_ids_88%>%group_by(subplot)%>%summarise(n_indiv=n())
```

```
## # A tibble: 6 x 2
##   subplot n_indiv
##     <dbl>   <int>
## 1       1      33
## 2       2      23
## 3       3      32
## 4       4      28
## 5       5      21
## 6       6      32
```

```
data_id_flowers_88%>%group_by(subplot)%>%summarise(n_indiv=n_distinct(id))
```

```
## # A tibble: 6 x 2
##   subplot n_indiv
##     <dbl>   <int>
## 1       1      33
## 2       2      23
## 3       3      32
## 4       4      28
## 5       5      21
## 6       6      32
```

```
data_ids_89%>%group_by(subplot)%>%summarise(n_indiv=n())
```

```
## # A tibble: 3 x 2
##   subplot n_indiv
##     <dbl>   <int>
```

```
## 1       1      38
## 2       2      15
## 3       3      42
```

```
data_id_flowers_89%>%group_by(subplot)%>%summarise(n_indiv=n_distinct(id))
```

```
## # A tibble: 3 x 2
##   subplot n_indiv
##     <dbl>   <int>
## 1       1      38
## 2       2      15
## 3       3      42
```

Yes, it matches.

See if the id values match between individual data and individual flower data.

```
unique(anti_join(data_id_flowers_87, data_ids_87, by = "id")$id)
```

```
## character(0)
```

```
# Show values of id from data_id_flowers_87 that are not in data_ids_87
unique(anti_join(data_id_flowers_88, data_ids_88, by = "id")$id)
```

```
## character(0)
```

```
# Show values of id from data_id_flowers_88 that are not in data_ids_88
unique(anti_join(data_id_flowers_89, data_ids_89, by = "id")$id)
```

```
## character(0)
```

```
# Show values of id from data_id_flowers_89 that are not in data_ids_89
```

Yes, they match.

Recalculate moments

I have recalculated all moments to check that everything matches with Johan's data for individuals (I might remove other moments later and keep only new versions of skewness and kurtosis).

```
moments_87 <- data_id_flowers_87 %>%
  group_by(id) %>%
  summarise(avFD_a=mean(opening_date),FFD_a=min(opening_date),
            MFD_a=median(opening_date), # Calculate also median
            LFD_a=max(opening_date),SD_a=sd(opening_date),
            var_a=var(opening_date), # Calculate also variance
            skew_a=ifelse(n()>2,skewness(opening_date),NA),
            kurt_a=ifelse(n()>2,kurtosis(opening_date),NA),
            # Calculate skewness and kurtosis when n_fl>2
            dur_a=LFD_a-FFD_a) %>%
```

```
  mutate(year=as.factor(1987))
moments_88 <- data_id_flowers_88 %>%
  group_by(id) %>%
  summarise(avFD_a=mean(opening_date),FFD_a=min(opening_date),
            MFD_a=median(opening_date), # Calculate also median
            LFD_a=max(opening_date),SD_a=sd(opening_date),
            var_a=var(opening_date), # Calculate also variance
            skew_a=ifelse(n()>2,skewness(opening_date),NA),
            kurt_a=ifelse(n()>2,kurtosis(opening_date),NA),
            # Calculate skewness and kurtosis when n_fl>2
            dur_a=LFD_a-FFD_a) %>%
  mutate(year=as.factor(1988))
moments_89 <- data_id_flowers_89 %>%
  group_by(id) %>%
  summarise(avFD_a=mean(opening_date),FFD_a=min(opening_date),
            MFD_a=median(opening_date), # Calculate also median
            LFD_a=max(opening_date),SD_a=sd(opening_date),
            var_a=var(opening_date), # Calculate also variance
            skew_a=ifelse(n()>2,skewness(opening_date),NA),
            kurt_a=ifelse(n()>2,kurtosis(opening_date),NA),
            # Calculate skewness and kurtosis when n_fl>2
            dur_a=LFD_a-FFD_a) %>%
  mutate(year=as.factor(1989))
moments <- full_join(full_join(moments_87,moments_88),moments_89)
```

Merge Johan's data for individuals for the 3 years

```
data_ids_87 <- data_ids_87 %>%
  mutate(year = as.integer(1987))
data_ids_88 <- data_ids_88 %>%
  mutate(year = as.integer(1988))
data_ids_89 <- data_ids_89 %>%
  mutate(year = as.integer(1989))
data_ids <- full_join(full_join(data_ids_87,data_ids_88),data_ids_89)
data_ids <- data_ids %>% mutate(year = as.factor(year))
```

Merge with my calculated moments

```
data_ids <- full_join(data_ids, moments)
```

Compare values of moments between Johan's calculations and mine

In how many ids are my calculations different from Johan's?

```r
nrow(data_ids %>% filter(!near(avFD_a,avFD)) %>%
       # Using near() to avoid small differences in decimals
       select(year,number,subplot,id,avFD,avFD_a))
```

```
## [1] 0
```

```r
# None after editing data
nrow(data_ids %>% filter(!near(FFD_a,FFD))%>%
       select(year,number,subplot,id,FFD,FFD_a))
```

```
## [1] 0
```

```r
# None after editing data
nrow(data_ids %>% filter(!near(LFD_a,LFD))%>%
       select(year,number,subplot,id,LFD,LFD_a))
```

```
## [1] 0
```

```r
# None after editing data
nrow(data_ids %>% filter(!near(SD_a,SD))%>%
       select(year,number,subplot,id,SD,SD_a))
```

```
## [1] 0
```

```r
# None after editing data
nrow(data_ids %>% filter(!near(skew_a,skew))%>%
       select(year,number,subplot,id,skew,skew_a))
```

```
## [1] 415
```

```r
# 415 rows are different
nrow(data_ids %>% filter(!near(kurt_a,kurt))%>%
       select(year,number,subplot,id,kurt,kurt_a))
```

```
## [1] 377
```

```r
# 377 rows are different
nrow(data_ids %>% filter(!near(dur_a,dur))%>%
       select(year,number,subplot,id,dur,dur_a))
```

```
## [1] 0
```

```r
# None after editing data
```

All moments have the same values except for skewness and kurtosis.

The skewness function that I used (from the moments package) calculates g1, the skewness of a sample based on the third moment of the data divided by the cube root of the second moment of the data, using the formula:

g1=(sum((X - mean(X))^3)/n)/(sum((X - mean(X))$^{2)/n)}$(3/2)

This is the formula for sample skewness, also known as Pearson's moment coefficient of skewness.

Excel uses the adjusted Fisher–Pearson standardized moment coefficient G1:

G1=(sqrt(n()*(n()-1))/(n()-2)*)g1

The kurtosis function that I used (from the moments package) calculates Pearson's measure of kurtosis:

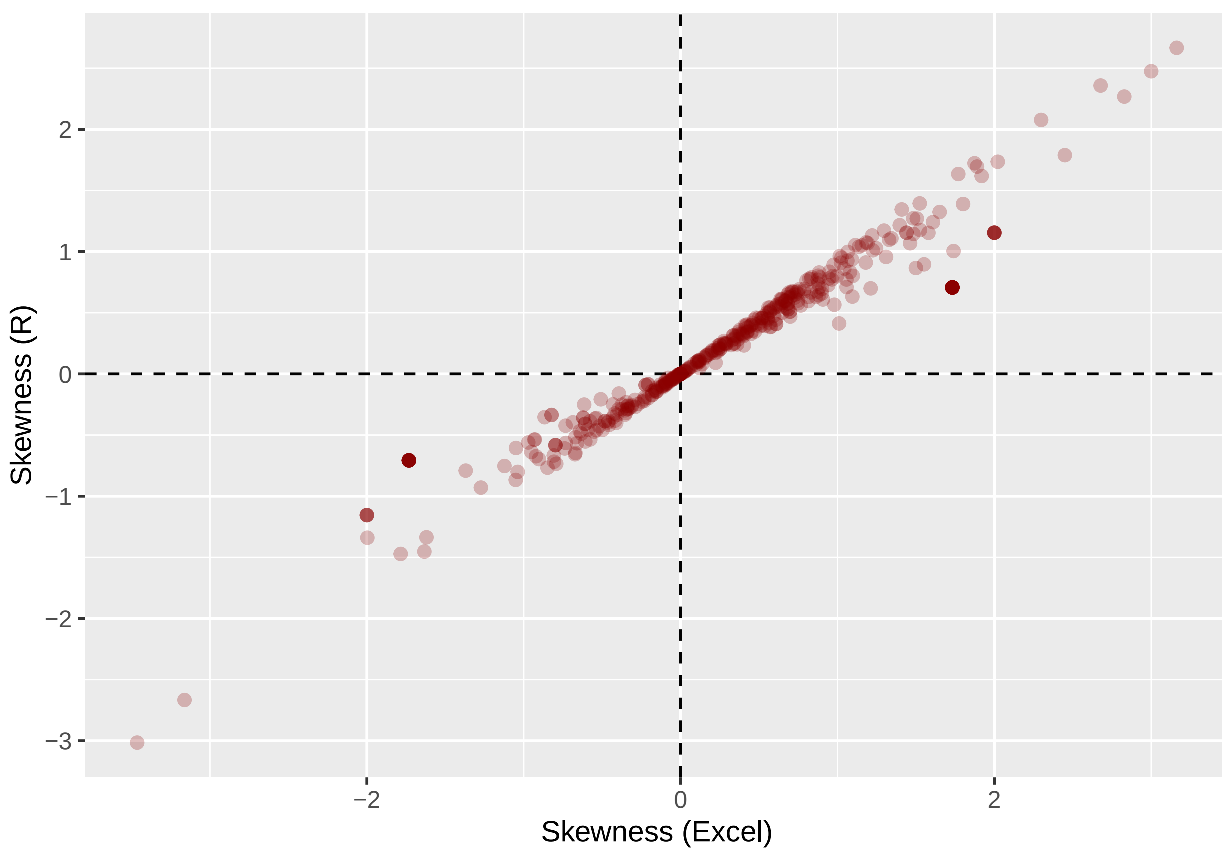
n*(sum((X-mean(X))$^{4))/((\text{sum}((X-\text{mean}(X))}$2)^2))

Excel uses:

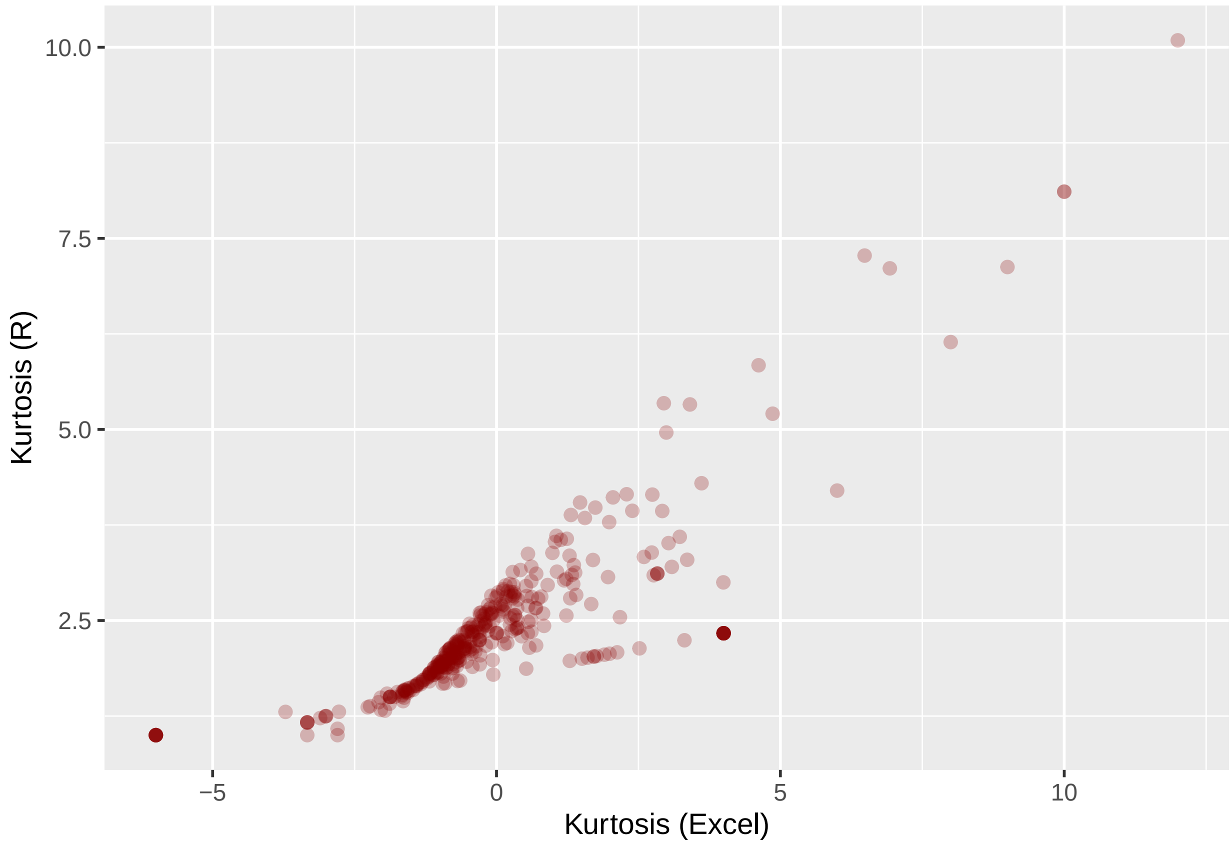
((n*(n+1))/((n-1)*(n-2)*(n-3))*sum(((X-mean(X))/sd(X))^4))-((3*((n-1)^2))/((n-2)*(n-3)))

Plots of skewness and kurtosis

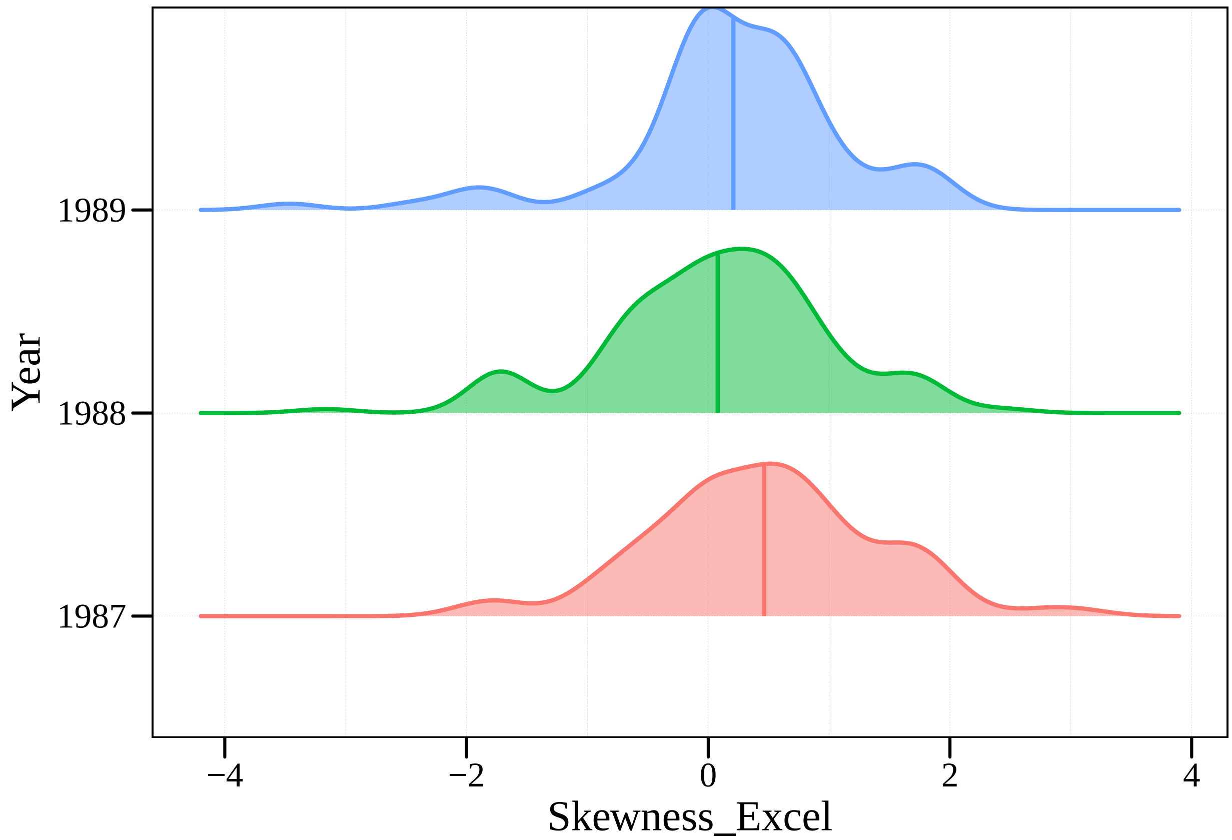
```
ggplot(data_ids,aes(x=skew,y=skew_a))+
  geom_vline(xintercept=0,linetype=2)+geom_hline(yintercept=0,linetype=2)+
  geom_point(shape=20,size=3,alpha=0.25,color="darkred")+
  xlab("Skewness (Excel)")+ylab("Skewness (R)")
```



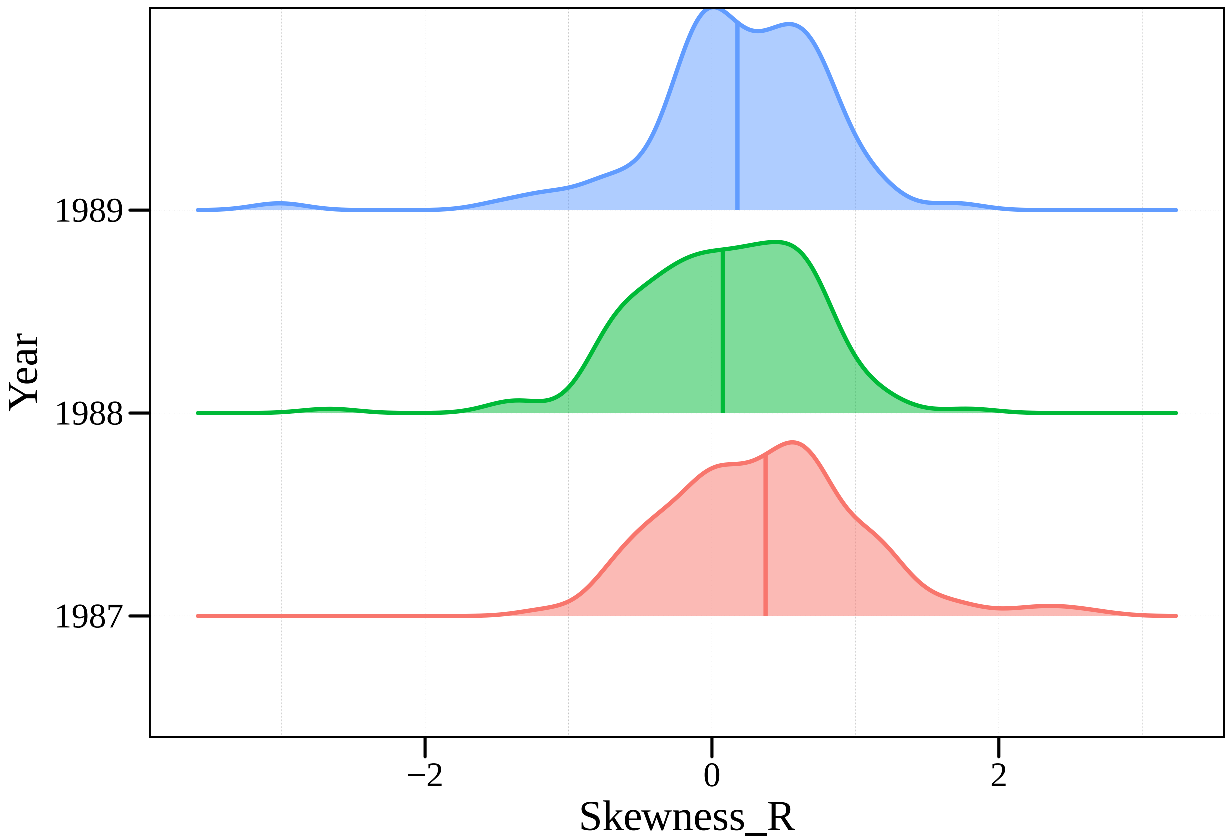

```
ggplot(data_ids,aes(x=kurt,y=kurt_a))+
  geom_point(shape=20,size=3,alpha=0.25,color="darkred")+
  xlab("Kurtosis (Excel)")+ylab("Kurtosis (R)")
```



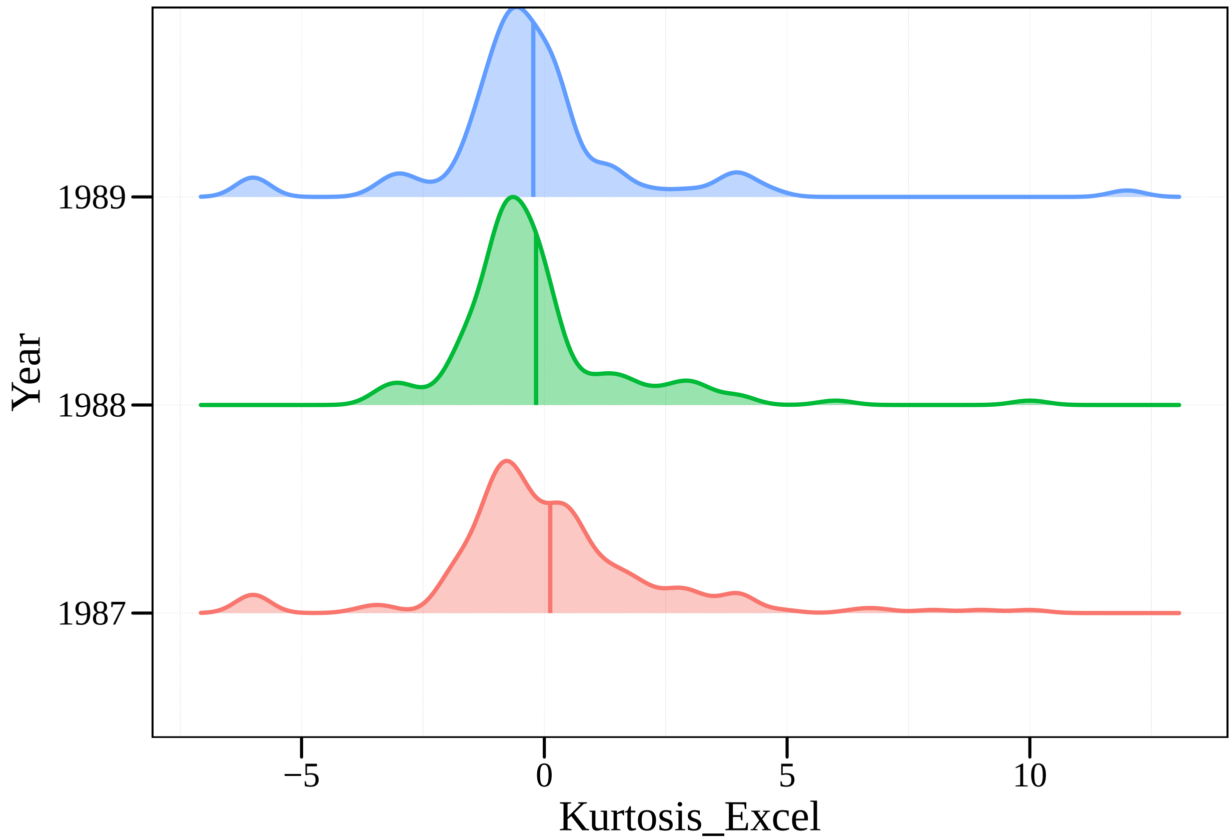

```
ggplot(data_ids,aes(x=skew,y=year,fill=year,color=year))+
  my_theme()+
  theme(panel.grid = element_line(color="grey",size=0.1,linetype=3))+
  labs(x="Skewness_Excel",y="Year")+
  geom_density_ridges(alpha=.5,scale=1,quantile_lines=TRUE,
                      quantile_fun=function(x,...)mean(x),size=0.75)
```



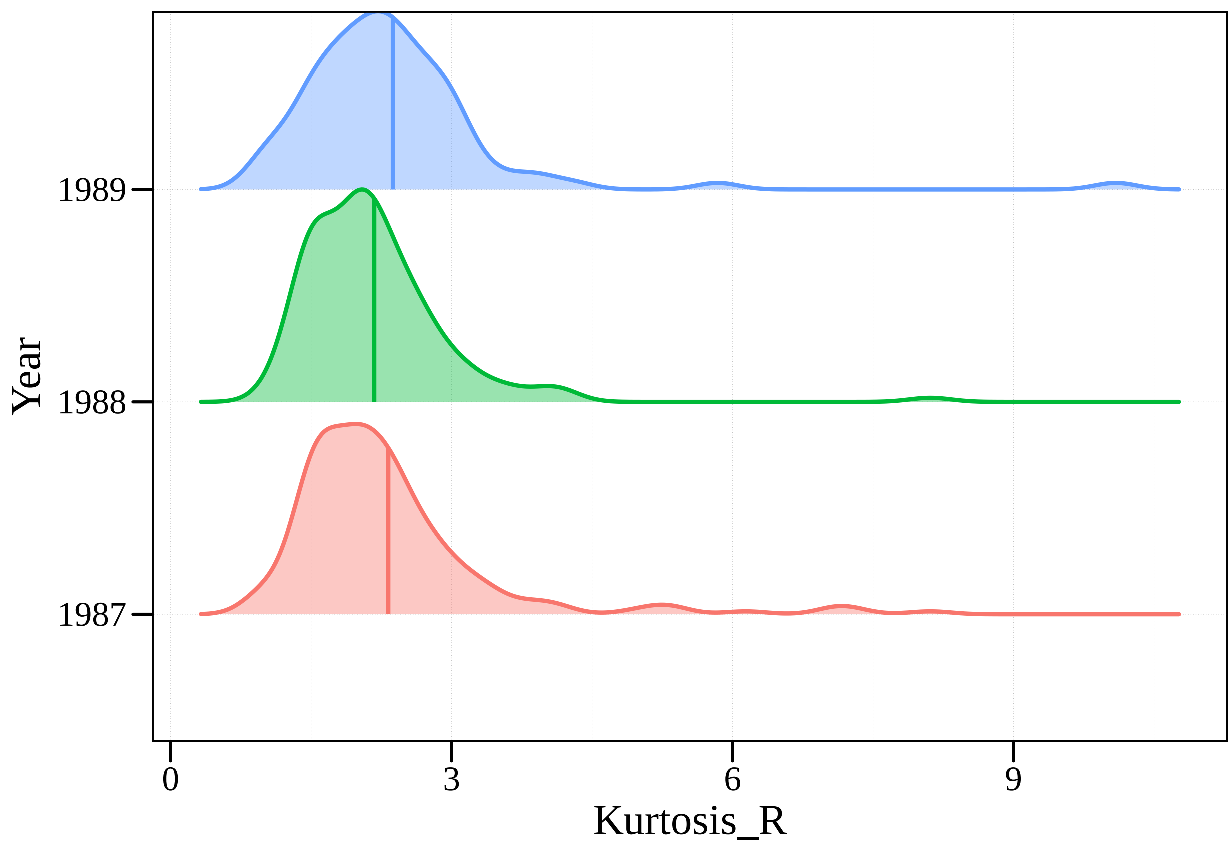

```
ggplot(data_ids,aes(x=skew_a,y=year,fill=year,color=year))+
  my_theme()+
  theme(panel.grid = element_line(color="grey",size=0.1,linetype=3))+
  labs(x="Skewness_R",y="Year")+
  geom_density_ridges(alpha=.5,scale=1,quantile_lines=TRUE,
                      quantile_fun=function(x,...)mean(x),size=0.75)
```




```
ggplot(data_ids,aes(x=kurt,y=year,fill=year,color=year))+
  my_theme()+
  theme(panel.grid = element_line(color="grey",size=0.1,linetype=3))+
  labs(x="Kurtosis_Excel",y="Year")+
  geom_density_ridges(alpha=.4,scale=1,quantile_lines=TRUE,
                      quantile_fun=function(x,...)mean(x),size=0.75)
```




```r
ggplot(data_ids,aes(x=kurt_a,y=year,fill=year,color=year))+
  my_theme()+
  theme(panel.grid = element_line(color="grey",size=0.1,linetype=3))+
  labs(x="Kurtosis_R",y="Year")+
  geom_density_ridges(alpha=.4,scale=1,quantile_lines=TRUE,
                      quantile_fun=function(x,...)mean(x),size=0.75)
```



So far I will keep only my calculated moments

Keep only my calculated moments

```
data_ids<-data_ids%>%
  select(number,subplot,id,n_fl,n_fr,fr_init,n_seed,n_preyed_seed,
                 fitness,imp_seed_preyed,n_seed_per_fr,prop_seed_preyed,year,
                 avFD_a,FFD_a,MFD_a,LFD_a,SD_a,var_a,skew_a,kurt_a,dur_a)%>%
  rename_at(vars(ends_with("_a")), ~sub("_a","",.))
```

Transform dates

The dates are given in terms of four- or five-day intervals after the first recording. Convert them to calendar dates, then to julian dates, and then to number of days after the vernal equinox.

First create a table with information on each date for each year.

```
dates <- tibble(year = as.factor(
  c(rep("1987",10),rep("1988",10),rep("1989",10))),
                date_num = rep(1:10,3), # Date in numeric format given in data
                date_calendar = as.Date(
                  ifelse(year==1987, seq(as.Date("1987-05-18"),
```

```r
                                       by = 4, length.out = 10),
                          # 1987: Start 18 May, 4-day intervals
                          ifelse(year==1988, seq(as.Date("1988-05-15"),
                                                 by = 5, length.out = 10),
                                 # 1988: Start 15 May, 5-day intervals
                                 seq(as.Date("1989-05-07"),
                                     by = 5, length.out = 10))),
                   # 1989: Start 7 May, 5-day intervals
                   # Calendar date
                   origin = "1970-01-01"),
  date_julian = yday(date_calendar), # Julian date
  date_vernal = ifelse(year==1987,date_calendar-as.Date("1987-03-21"),
                       ifelse(year==1988,date_calendar-as.Date("1988-03-20"),
                              date_calendar-as.Date("1989-03-20"))))
# Days after vernal equinox
# Data on vernal equinox dates from https://data.giss.nasa.gov/ar5/srvernal.html
```

Calculate, for each year, the intercept and slope of the relationship among date_num (x) and date_vernal or date_calendar (y).

```r
dates <- dates %>%
  group_by(year) %>%
  mutate(date_intercept_v = ifelse(year==1987,min(date_vernal)-4,
                                   min(date_vernal)-5),
         date_intercept_c = as.Date(ifelse(year==1987,min(date_calendar)-4,
                                   min(date_calendar)-5),origin = "1970-01-01"),
         date_slope_v = (date_vernal-date_intercept_v)/date_num,
         date_slope_c = (date_calendar-date_intercept_c)/date_num)
dates_summary <- summarise(dates,
                           date_intercept_v = mean(date_intercept_v),
                           date_intercept_c = mean(date_intercept_c),
                           date_slope_v = mean(date_slope_v),
                           date_slope_c = mean(date_slope_c))
```

Transform avFD, FFD, MFD and LFD to calendar dates (avFFD_c, FFD_c, MFD_c, LFD_c) and to days after vernal equinox (avFFD_v, FFD_v, MFD_v, LFD_v).

```r
data_ids <- data_ids %>%
  left_join(dates_summary, by = c("year" = "year")) %>%
  mutate(avFD_c = date_slope_c * avFD + date_intercept_c,
         FFD_c = date_slope_c * FFD + date_intercept_c,
         MFD_c = date_slope_c * MFD + date_intercept_c,
         LFD_c = date_slope_c * LFD + date_intercept_c,
         avFD_v = date_slope_v * avFD + date_intercept_v,
         FFD_v = date_slope_v * FFD + date_intercept_v,
         MFD_v = date_slope_v * MFD + date_intercept_v,
         LFD_v = date_slope_v * LFD + date_intercept_v)
```

Standardize traits and relativize fitness within years

```
data_ids<-data_ids%>%
  group_by(year)%>%
  mutate(across(c(n_fl,avFD:dur), scale, .names = "{col}_std"))%>%
  mutate(across(c(n_fl_std:dur_std),as.vector))%>%
  mutate(fitness_rel = fitness / mean(fitness))%>%
  ungroup()
# When standardizing, we get the same result for FFD, MFD and LFD
# than for FFD_v, MFD_v and LFD_v, so I used the first
```

Save clean data as .csv

```
write_csv(data_ids,"data/clean/data_ids.csv")
```

Session info

```
sessionInfo()
```

```
## R version 4.2.2 (2022-10-31 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 22621)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.utf8
## [2] LC_CTYPE=English_United States.utf8
## [3] LC_MONETARY=English_United States.utf8
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.utf8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] ggridges_0.5.4     ggthemes_4.2.4     RColorBrewer_1.1-3 moments_0.14.1
##  [5] lubridate_1.9.0    timechange_0.2.0   readxl_1.4.1       forcats_0.5.2
##  [9] stringr_1.5.0      dplyr_1.0.10       purrr_1.0.1        readr_2.1.3
## [13] tidyr_1.2.1        tibble_3.1.8       ggplot2_3.4.0      tidyverse_1.3.2
##
## loaded via a namespace (and not attached):
##  [1] assertthat_0.2.1    digest_0.6.31       utf8_1.2.2
##  [4] R6_2.5.1            cellranger_1.1.0    backports_1.4.1
##  [7] reprex_2.0.2        evaluate_0.20       highr_0.10
## [10] httr_1.4.4          pillar_1.8.1        rlang_1.0.6
```

```
## [13] googlesheets4_1.0.1 rstudioapi_0.14     rmarkdown_2.19
## [16] labeling_0.4.2      googledrive_2.0.0   bit_4.0.5
## [19] munsell_0.5.0       broom_1.0.2         compiler_4.2.2
## [22] modelr_0.1.10       xfun_0.36           pkgconfig_2.0.3
## [25] htmltools_0.5.4     tidyselect_1.2.0    fansi_1.0.3
## [28] crayon_1.5.2        tzdb_0.3.0          dbplyr_2.3.0
## [31] withr_2.5.0         grid_4.2.2          jsonlite_1.8.4
## [34] gtable_0.3.1        lifecycle_1.0.3     DBI_1.1.3
## [37] magrittr_2.0.3      scales_1.2.1        vroom_1.6.0
## [40] cli_3.6.0           stringi_1.7.12      farver_2.1.1
## [43] fs_1.5.2            xml2_1.3.3          ellipsis_0.3.2
## [46] generics_0.1.3      vctrs_0.5.1         tools_4.2.2
## [49] bit64_4.0.5         glue_1.6.2          hms_1.1.2
## [52] parallel_4.2.2      fastmap_1.1.0       yaml_2.3.6
## [55] colorspace_2.0-3    gargle_1.2.1        rvest_1.0.3
## [58] knitr_1.41          haven_2.5.1
```