

Parturition Date Calculation

Andrew McAdam

November 26, 2024

Set-up

Load packages

Connect to Database

```
con <- krsp_connect (host = "krsp.cepb5cjvqban.us-east-2.rds.amazonaws.com",
                     dbname = "krsp",
                     username = Sys.getenv("krsp_user"),
                     password = Sys.getenv("krsp_password")
                     )
con2 <- krsp_connect (host = "krsp.cepb5cjvqban.us-east-2.rds.amazonaws.com",
                      dbname = "krsp2024",
                      username = Sys.getenv("krsp_user"),
                      password = Sys.getenv("krsp_password")
                      )
```

Rationale

We need a transparent and consistent system for calculating Parturition Dates based on our data.

Notes

- 1) This code generates a csv file that has calculated parturition dates for first litters of the year only.
- 2) The algorithm seems to be working fine based on my checks but this is a good opportunity to do more error checking.
- 3) The csv includes a 'part_source' variable that indicates how the parturition date was calculated. This can be used to exclude if people do not want to use data from a particular method. 'field' is the only one that could be potentially suspect because this is the method that cannot be entirely recreated from other data.
- 4) There are two scripts to clean errors. One can be excluded once the annual data cleanup is complete. The other should be kept but does not need to be updated because it is for older years only. It should not apply to more recent years.

What is Birth Weight

First need to know what birth weights are, whether they depend on food availability and litter size. This was done previously for a separate analysis where we knew birth dates of pups very closely (very few days between first lactating and last pregnant). This is not included below, but it works out to be about 8g birth weight and about 1.8g/day of growth. Birth weight doesn't seem to be affected by litter size and food. We can probably use these metrics to calculate parturition dates based on weights. We can adjust this as needed.

Import Data

```
grids <- c("KL", "SU", "CH", "JO", "LL", "AG")

trapping<- tbl(con, "trapping") %>%
  rename(grid = gr) %>%
  # restrict to grids of interest
  filter(grid %in% grids) %>%
  # remove by catch records
  filter(!is.na(squirrel_id),
         !(ft %in% c(91, 92, 93, 94)),
         !is.na(date)) %>%
  mutate(year = year(date)) %>%
  filter(year>2012) %>% # rep_con was not correct prior to 2013. See dbatrapping below
  collect(n = Inf) %>%
  mutate(squirrel_id = as.integer(squirrel_id),
         date = ymd(date))

trapping_2024<- tbl(con2, "trapping") %>%
  rename(grid = gr) %>%
  # restrict to grids of interest
  filter(grid %in% grids) %>%
  # remove by catch records
  filter(!is.na(squirrel_id),
         !(ft %in% c(91, 92, 93, 94)),
         !is.na(date)) %>%
  mutate(year = year(date)) %>%
  filter(year>2012) %>% # rep_con was not correct prior to 2013. See dbatrapping below
  collect(n = Inf) %>%
  mutate(squirrel_id = as.integer(squirrel_id),
         date = ymd(date))

trapping<-rbind(trapping, trapping_2024)

dbatrapping<- tbl(con, "dbatrapping") %>%
  rename(grid = gr) %>%
  # restrict to grids of interest
  filter(grid %in% grids) %>%
  # remove by catch records
  filter(!is.na(squirrel_id),
         !(ft %in% c(91, 92, 93, 94)),
         !is.na(date)) %>%
  mutate(year = year(date)) %>%
```

```

filter(year<2013) %>%
collect(n = Inf) %>%
mutate(squirrel_id = as.integer(squirrel_id),
       date = ymd(date))

litter<- tbl(con, "litter") %>%
select(litter_id=id, year=yr, squirrel_id, fieldBDate, date1, tagDt, br, grid, ln) %>%
collect() %>%
mutate (date1 = ymd(date1),
       tagDt = ymd(tagDt),
       fieldBDate = ymd(fieldBDate),
       ln = as.numeric(ln))

litter_2024<- tbl(con2, "litter") %>%
select(litter_id=id, year=yr, squirrel_id, fieldBDate, date1, tagDt, br, grid, ln) %>%
collect() %>%
mutate (date1 = ymd(date1),
       tagDt = ymd(tagDt),
       fieldBDate = ymd(fieldBDate),
       ln = as.numeric(ln))

litter<-rbind(litter, litter_2024)

juvenile<- tbl(con, "juvenile") %>%
select(squirrel_id, litter_id, sex, weight, tagWT) %>%
collect()

juvenile_2024<- tbl(con2, "juvenile") %>%
select(squirrel_id, litter_id, sex, weight, tagWT) %>%
collect()

juvenile<-rbind(juvenile, juvenile_2024)

```

Correct Errors in the Data

I wrote two scripts to correct errors that I found in the data. These are in two separate files. One file contains errors in the database that need to be corrected. Once these are corrected in the next database cleanup, this file would presumably not be needed anymore. The second file contains situations that aren't really errors in the database, but they are bits of data that cause the algorithm to calculate bad part dates.

As an example of the latter, in the old trapping table (dbtrapping) we had a "BRSTATUS 4 = LAC – breeder – elongated, generally more swollen than 3 (may be lactating)". In the code, I included this as a possible indication that a female might have been lactating. This was helpful for finding first lactating dates. But you can see from the protocol that BRSTATUS==4 does not necessarily mean that the female was lactating. So in cases where this resulted in odd parturition dates I suppressed these observations (i.e. temporarily deleted them) from the calculation. Other cases involved situations where there was likely a bad assessment of reproductive status or nipples. I had enough confidence in the problem to suppress the observation, but not enough to warrant deleting it from the database (i.e. a database change).

```

# run code to correct database errors. This can be eliminated following the next database
# update if these errors are cleaned in the database.
source("partdate_database_corrections.R")

```

```
# run code to correct administrative errors. These aren't errors in the database but just
# errors that cause problems for the algorithm
source("partdate_admin_data_corrections.R")
```

Calculate First Lactating Capture and Last Pregnant Capture

What is each female's first lactating capture in each year?

This code searches for the first lactating observation for a female in a given year. Lactation could be identified by reproductive status, nipple condition or by the first nest entry (date1). The coding is different for the older data prior to 2012 (dbatrapping) and the newer data since 2012 (trapping).

```
first_lac1<-trapping %>%
  filter(nipple %in%c(4, 5, 6), year>2012) %>%
  group_by(year, squirrel_id) %>%
  slice(which.min(date)) %>%
  select(year, squirrel_id, first_lac_date_temp=date, first_lac_wgt=wgt)

# Add in observations from dbatrapping
first_lac2<-dbatrapping %>%
  filter(BrStatus %in%c(4, 5, 6) | RC==3,
         year<2013) %>% # this is a bit problematic because BC==4 does not necessarily indicate
                        # lactation, but it was a helpful way to identify issues. After these
                        # are cleaned up the code should revert back to just using 5 or 6.
  # filter(BrStatus %in%c(5, 6) | RC==3,
  #        year<2013) %>%
  # filter(RC==3,
  #        year<2013) %>%
  group_by(year, squirrel_id) %>%
  slice(which.min(date)) %>%
  select(year, squirrel_id, first_lac_date_temp=date, first_lac_wgt=wgt)

first_lac<-rbind(first_lac1, first_lac2)

# Add in date1 from litter because this is a confirmation that the female is lactating even if
# she hasn't been captured lactating yet. Take the minimum date of the first lactating capture and
# date1 as the first evidence that she was lactating.
litter_temp<-litter %>%
  filter(grid %in% grids,
         ln==1,
         !is.na(date1),
         date1!=tagDt) %>% # also excludes situations where this the tag date because she could be
                        # pregnant again with the next litter
  select(year, squirrel_id, date1) %>%
  collect() %>%
  mutate (date1=ymd(date1))

first_lac<-first_lac %>%
  left_join(litter_temp, by=c("squirrel_id", "year"), multiple="all") %>%
  mutate(first_lac_date = min(first_lac_date_temp, date1, na.rm=T)) %>%
  select(year, squirrel_id, first_lac_date, first_lac_wgt)
```

When was the female's last pregnant capture?

This is also based on trapping, but only considers trapping records prior to the first lactating observation for that female in that year. Pregnancy was determined through reproductive condition (`rep_con > 2` in trapping; `RC == 2` or `4` in `dbatrapping`).

```
last_preg1<-trapping %>%
  select(year, squirrel_id, date, wgt, sex, rep_con, nipple) %>%
  inner_join (first_lac, by=c("squirrel_id", "year"), multiple="all") %>%
  filter(date<first_lac_date, rep_con>2, year>2012) %>%
  group_by(year, squirrel_id) %>%
  slice(which.max(date)) %>%
  rename(last_preg_date = date) %>%
  ungroup() %>%
  mutate(trap_part_date = if_else((first_lac_date-last_preg_date)<10, last_preg_date+round(0.5*(first_lac_date-last_preg_date)),
    first_lac_date)) %>%
  select(year, squirrel_id, last_preg_date, wgt, sex, rep_con, first_lac_date, first_lac_wgt, trap_part_date)

last_preg2<-dbatrapping %>%
  select(year, squirrel_id, date, wgt, sex=Sex, RC) %>%
  inner_join (first_lac, by=c("squirrel_id", "year")) %>%
  filter(date<first_lac_date, RC %in% c(2,4), year<2013) %>%
  group_by(year, squirrel_id) %>%
  slice(which.max(date)) %>%
  rename(last_preg_date = date) %>%
  ungroup() %>%
  mutate(trap_part_date = if_else((first_lac_date-last_preg_date)<10, last_preg_date+round(0.5*(first_lac_date-last_preg_date)),
    rep_con = NA) %>%
  select(year, squirrel_id, last_preg_date, wgt, sex, rep_con, first_lac_date, first_lac_wgt, trap_part_date)

last_preg<-rbind(last_preg1, last_preg2) %>%
  mutate (wgt = ifelse(wgt == 0, NA, wgt),
    first_lac_wgt = ifelse(first_lac_wgt == 0, NA, first_lac_wgt),
    wgt_drop = wgt - first_lac_wgt,
    day_diff = as.numeric(first_lac_date - last_preg_date))

# exclude cases where the female did not drop more than 5g in weight.
# This will hopefully exclude cases where a female was listed as lactating when she was
# still pregnant and some cases where she was missed as lactating for a while and still called pregnant

last_preg<-last_preg %>%
  filter(wgt_drop > 5)
```

Calculate Parturition dates based on trapping data

The parturition date from trapping (above) is calculated as median date between the last pregnant observation and the first lactating observation. This is only calculated if there is less than 10 days between these two dates. This is only calculated for situations where the female dropped more than 5g between pregnancy and lactation.

Calculate Parturition dates based on pup weights

This is the predicted parturition date based on the weight of the pups at first nest entry. Working with the litter table, moving forward we consider only first litters and exclude those where females were pregnant but never gave birth ($br==7$). The average weight of the pups was used to calculate when we think they might have been born. This was based on a birth weight of 8g and a growth rate of 1.8g/day. This date was further restricted to be before the first lactating observation and after the last pregnancy observation. The weight of pups at nest1 was only used to predict the parturition date when the average weight of the pups was less than 50g.

```
litter<-litter %>%
  filter(grid%in% grids,
         ln==1,
         br!=7) %>% #should not include litters where females were pregnant but never gave birth
  left_join(last_preg, by=c("year", "squirrel_id")) %>%
  mutate(age1_trap = date1-trap_part_date,
         age1_field = date1-fieldBDate,
         grid=as.factor(grid))

juvenile<-juvenile %>%
  mutate(weight = ifelse(weight == 0, NA, weight),
         tagWT = ifelse(tagWT == 0, NA, tagWT))

avg_wts<-juvenile %>%
  group_by(litter_id) %>%
  summarize(avg_weight = mean(weight),
         avg_tagWT = mean(tagWT))

litter<-litter %>%
  left_join(avg_wts, by = "litter_id") %>%
  mutate(weight_part_date2 = date1 - ((avg_weight-8)/1.8),
         j_last_preg_date = yday(last_preg_date),
         j_first_lac_date = yday(first_lac_date),
         j_fieldBDate = yday(fieldBDate),
         j_trap_part_date = yday(trap_part_date))

litter<-litter %>%
  mutate(weight_part_date2 = if_else(avg_weight < 50, weight_part_date2, as.Date(NA)))
#do not calculate part date using weights when pups are greater than 50g

litter<-litter %>%
  mutate(j_weight_part_date2 = yday(weight_part_date2))

# Because the last pregnant date and the first lactating date are used as limiters on the
# parturition date below then I am going to replace missing data for these with a very early
# and very early date so that these limiters do not throw NAs when missing

litter<-litter %>%
  mutate (last_preg_date = if_else(is.na(last_preg_date), ymd("1980-01-01"), last_preg_date),
         first_lac_date = if_else(is.na(first_lac_date), ymd("2080-01-01"), first_lac_date))
```

Parturition Date Definition

This is now our working definition of how we calculate parturition date:

- Parturition date is defined to be... median date between first lactating and last pregnant if there is less than 10 days between these. Otherwise it is the weight-based part date (assumes a birth weight of 8g and a growth rate of 1.8g/day) with the constraint that the weight-based part date needs to fall between the last pregnant capture and the first lactating capture. Weights are only used to calculate parturition dates when pup weights are less than 50g.

```
litter<-litter %>%
  mutate(j_calculated_part_date = ifelse(weight_part_date2 > first_lac_date, j_first_lac_date,
                                         ifelse(weight_part_date2 < last_preg_date, j_last_preg_date, .
                                         # makes sure that the weight birth date is between the first lac and the last preg

litter<-litter %>%
  mutate (j_calculated_part_date2 = coalesce (j_trap_part_date, j_calculated_part_date, j_fieldBDate),
          part_source = ifelse(!is.na(j_trap_part_date), "trap", ifelse(!is.na(j_calculated_part_date),
          part_source = as.factor(part_source))
```

Corrections

In some cases the trapping data really don't match the weights of the pups at first nest entry. In some cases the trapping data look to be clear about a parturition date but this cannot be possible based on the weights of the pups. It's unclear what is incorrect. In these cases the parturition dates were excluded. Many from 1990.

```
litter<-litter %>%
  filter (litter_id != 966,
          litter_id != 653,
          litter_id != 746,
          litter_id != 732,
          litter_id != 741,
          litter_id != 626,
          litter_id != 628,
          litter_id != 542,
          litter_id != 924)

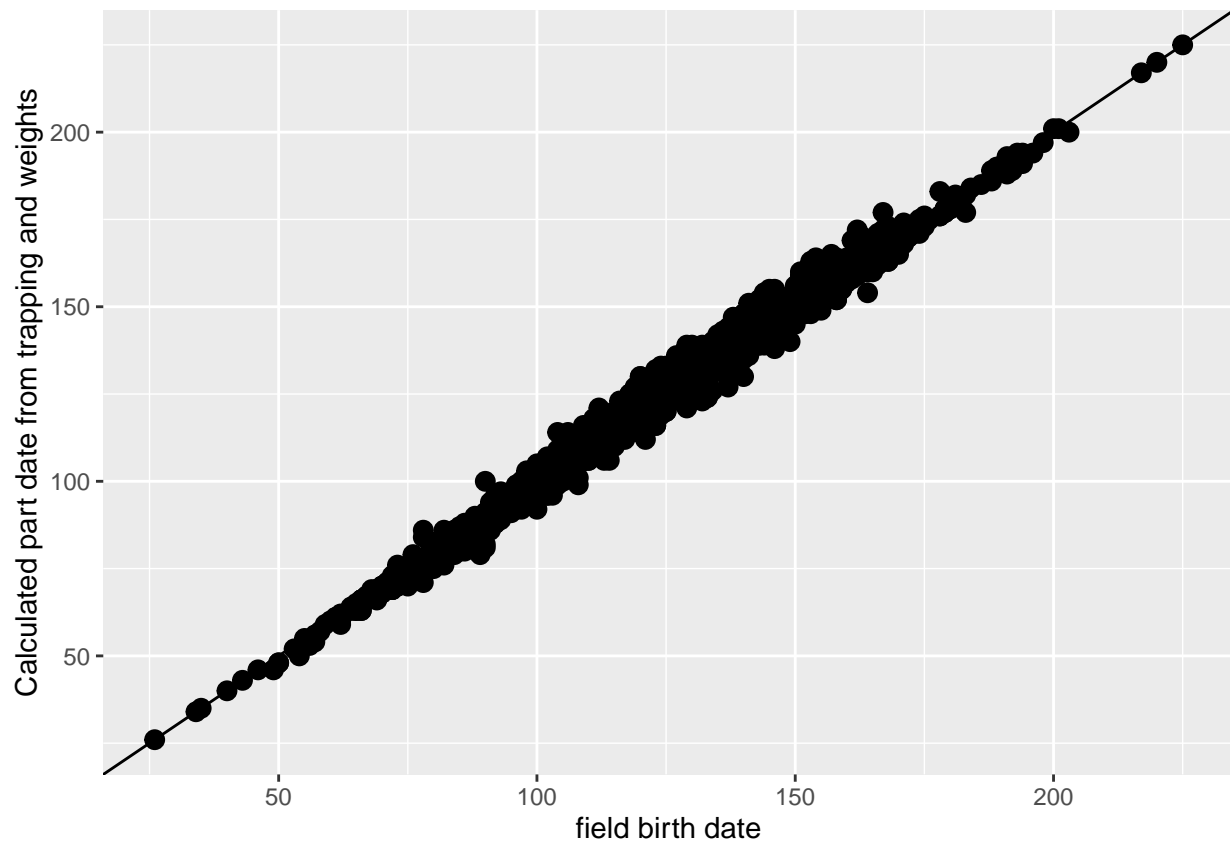
#924 is litter by unidentified mother
```

Check plots

This is a plot of the calculated parturition date plotted against the fieldBDate as entered by the crew in the field. A 1:1 line is fitted. Any observations with a discrepancy of >10 days have been sleuthed (there were initially about 200 of these). These were solved by improvements to the algorithm (e.g. the restriction that pup weights must be <50g to be used to predict the part date removed some impossible predictions). In other cases, these discrepancies revealed errors in the database (e.g. situations where the trapping data clearly indicated that the litter record was in fact the second litter of the year and not the first; typos in fieldBDate, date1, etc). These were checked using data from the database and going back to raw nest sheets.

I checked about a dozen cases where the discrepancy was 10 days and these were almost all simply cases that could be explained by slight differences in the calculation of birth dates from pup weights.

```
litter %>%
  filter(!is.na(j_calculated_part_date2), !is.na(j_fieldBDate)) %>%
  ggplot(aes(x=j_fieldBDate, y=j_calculated_part_date2)) +
    geom_point(size = 3) +
    geom_abline(slope=1, intercept=0) +
    ylab("Calculated part date from trapping and weights") +
    xlab("field birth date")
```



Clean and Write File

Simply the large file, rename the `part_date` variable and output the file as a `.csv`. This is what people should then import for future analyses.

```
litter<-litter %>%
  rename(part_date = j_calculated_part_date2) %>%
  select(litter_id, year, grid, squirrel_id, ln, part_date, part_source)

litter %>%
  write.csv("part_dates.csv")
```

Summaries

Tabular and graphical displays of the parturition date data.

Table of sample sizes for each grid in each year

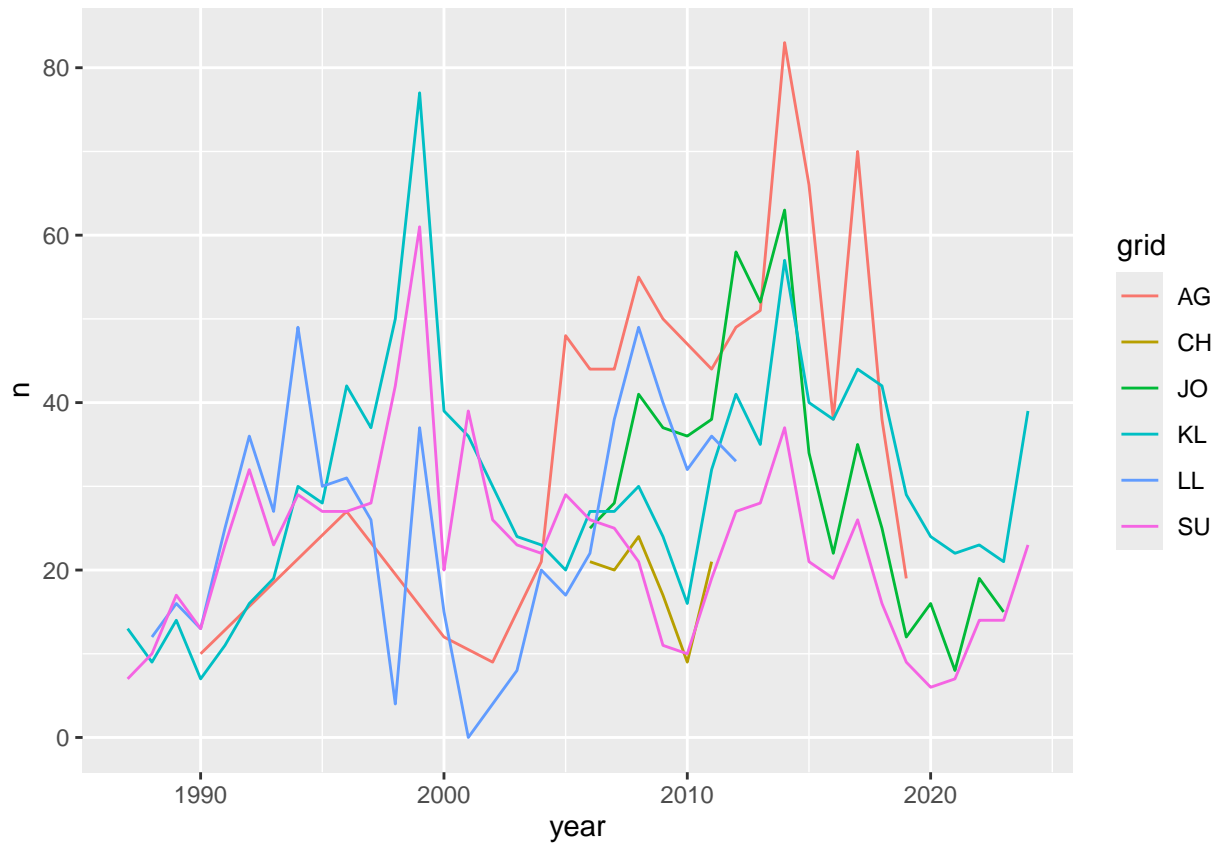
```
library(knitr)

litter %>%
  group_by(year, grid) %>%
  summarize(n=sum(!is.na(part_date))) %>%
  spread(grid, n, fill = NA) %>%
  kable()
```

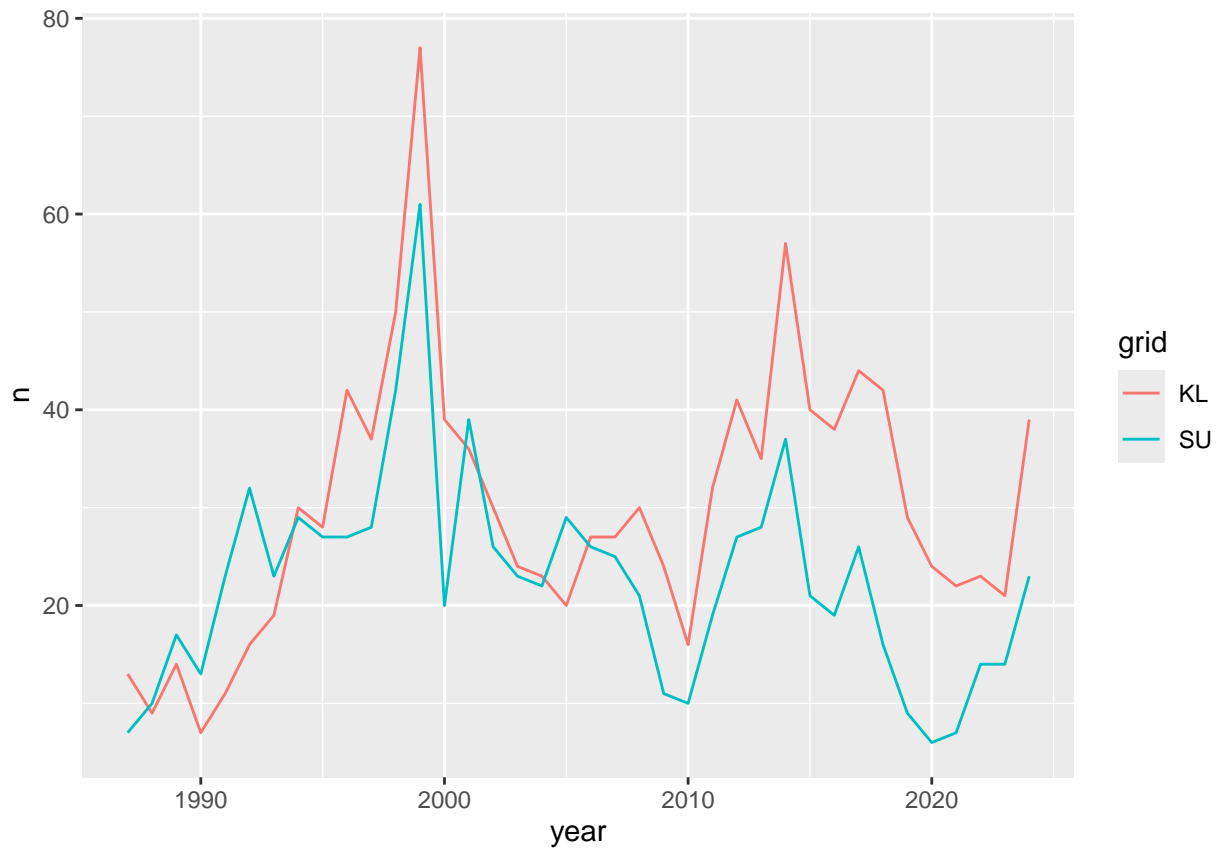
year	AG	CH	JO	KL	LL	SU
1987	NA	NA	NA	13	NA	7
1988	NA	NA	NA	9	12	10
1989	NA	NA	NA	14	16	17
1990	10	NA	NA	7	13	13
1991	NA	NA	NA	11	25	23
1992	NA	NA	NA	16	36	32
1993	NA	NA	NA	19	27	23
1994	NA	NA	NA	30	49	29
1995	NA	NA	NA	28	30	27
1996	27	NA	NA	42	31	27
1997	NA	NA	NA	37	26	28
1998	NA	NA	NA	50	4	42
1999	NA	NA	NA	77	37	61
2000	12	NA	NA	39	15	20
2001	NA	NA	NA	36	0	39
2002	9	NA	NA	30	NA	26
2003	NA	NA	NA	24	8	23
2004	21	NA	NA	23	20	22
2005	48	NA	NA	20	17	29
2006	44	21	25	27	22	26
2007	44	20	28	27	38	25
2008	55	24	41	30	49	21
2009	50	17	37	24	40	11
2010	47	9	36	16	32	10
2011	44	21	38	32	36	19
2012	49	NA	58	41	33	27
2013	51	NA	52	35	NA	28
2014	83	NA	63	57	NA	37
2015	66	NA	34	40	NA	21
2016	38	NA	22	38	NA	19
2017	70	NA	35	44	NA	26
2018	38	NA	25	42	NA	16
2019	19	NA	12	29	NA	9
2020	NA	NA	16	24	NA	6
2021	NA	NA	8	22	NA	7
2022	NA	NA	19	23	NA	14
2023	NA	NA	15	21	NA	14
2024	NA	NA	NA	39	NA	23

Plot of sample size for each grid by year.

```
litter %>%  
  group_by(year, grid) %>%  
  summarize(n=sum(!is.na(part_date))) %>%  
  ggplot(aes(x=year, y=n, color=grid))+  
  geom_line()
```



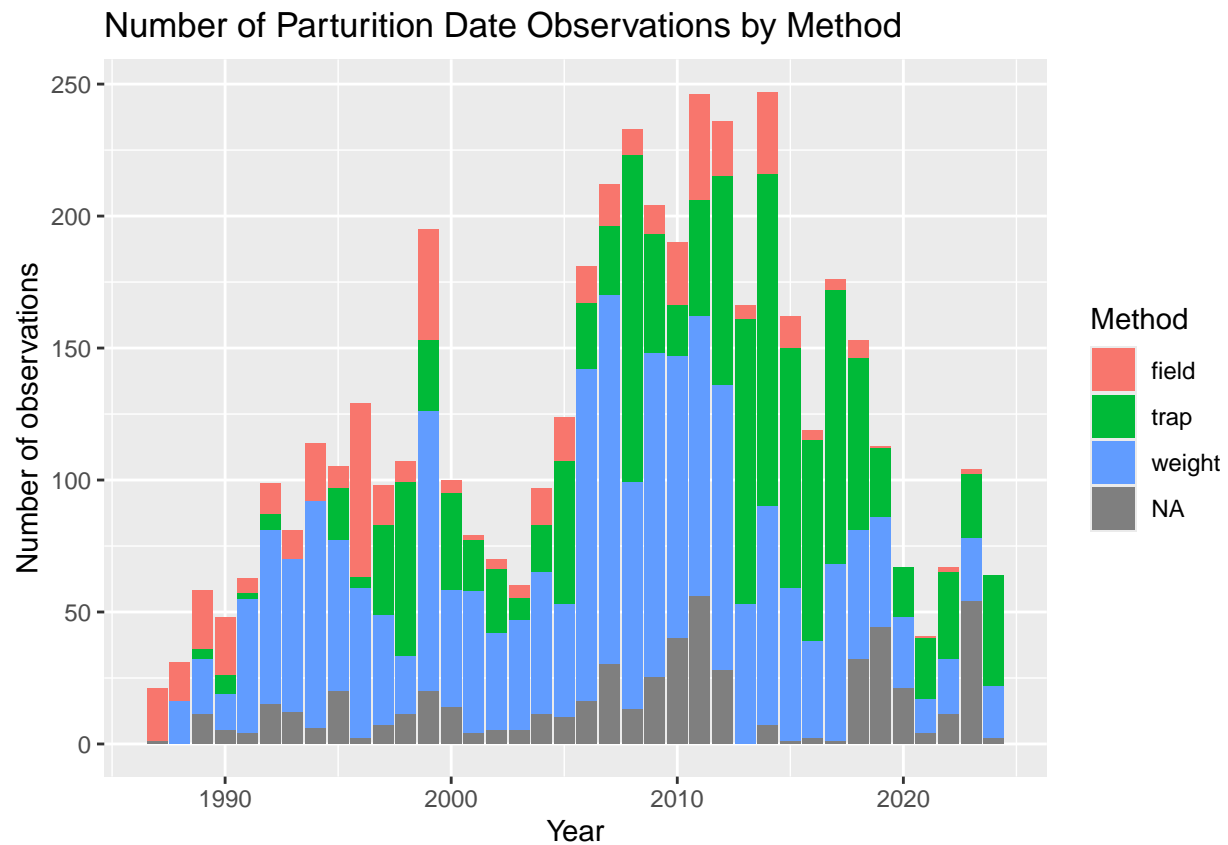
```
litter %>%  
  filter(grid %in% c("KL", "SU")) %>%  
  group_by(year, grid) %>%  
  summarize(n=sum(!is.na(part_date))) %>%  
  ggplot(aes(x=year, y=n, color=grid))+  
  geom_line()
```



Methods by Year

```
method_obs<-litter %>%
  group_by(year, part_source) %>%
  summarize(n=n())

ggplot(method_obs, aes(fill=part_source, y=n, x=year)) +
  geom_bar(position="stack", stat="identity")+
  ylab("Number of observations")+
  xlab("Year")+
  ggtitle("Number of Parturition Date Observations by Method")+
  scale_fill_discrete(name = "Method")
```

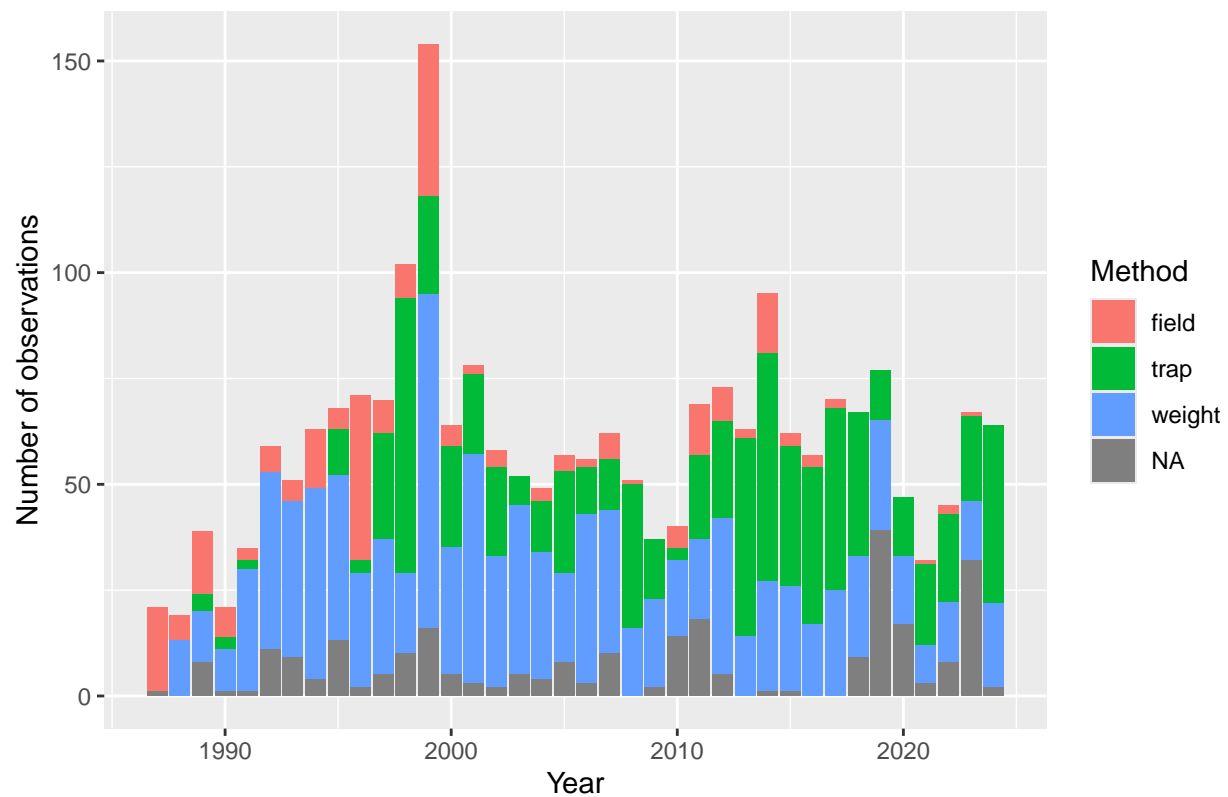


Methods by Year - KL and SU only

```
method_obs2<-litter %>%
  filter(grid%in% c("KL", "SU")) %>%
  group_by(year, part_source) %>%
  summarize(n=n())

ggplot(method_obs2, aes(fill=part_source, y=n, x=year)) +
  geom_bar(position="stack", stat="identity")+
  ylab("Number of observations")+
  xlab("Year")+
  ggtitle("Number of Parturition Date Observations by Method - KL and SU only")+
  scale_fill_discrete(name = "Method")
```

Number of Parturition Date Observations by Method – KL and SU only



Density plot of parturition dates by year for KL and SU

This is a ridgeline plot.

```
#####
library (ggridges)
library (viridis)
part_ridgeline_plot<-litter %>%
  filter(grid %in% c("KL", "SU")) %>%
  mutate(yrf=as.factor(year)) %>%
  ggplot(aes(x=part_date, y=yrf, fill = after_stat(x))) +
  geom_density_ridges_gradient(scale = 3, rel_min_height = 0.01) +
  scale_fill_viridis(name = "Julian", option = "C") +
  xlim(50, 200) +
  labs(title = 'Parturition Dates 1987 to 2024') +
  xlab("Parturition date (Julian day)") +
  ylab("")+
  theme(
    legend.position="none",
    panel.spacing = unit(0.1, "lines"),
    strip.text.x = element_text(size = 8)
  )

pdf(file="Part_Ridgeline.pdf",width=6, height=8)
print (part_ridgeline_plot)
dev.off()
```

```
## pdf
## 2
```

```
print (part_ridgeline_plot)
```

Parturition Dates 1987 to 2024

