The Best Cows

My sister called me the other day. She needed some help with formulas in excel. My sister is a cattle rancher in Montana. She manages a herd of a little over 100 cows pairs. She wanted to know which cows and sires (another word for bull cows) produce the heaviest young after one year. The heaviest young are important becuase many of her calfs are sold at auction after their first summer. The heavier they are the more money she can make.

Cleaning the dataset

My sister had done a really great job keeping track of life history information for her cows. But, a lot of her data was in different spreadsheets that used different column names.

Load the our libraries

We'll need the `tidyverse` packages for cleaning up the data and the `readxl` library to read the Excel file that I've combined the cow and calf information into. All of the data was in different spreadhseets that had a lot of formatting so the first step was to copy and paste all of the information into a single `.xlsx`.

For this step you can download the data here.

```
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------------- tidyverse 1
```

```
## v ggplot2 3.2.1     v purrr   0.3.2
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   0.8.3     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
```

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

```
library(readxl)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

```
d_2019_1<-read_xlsx("calf_info.xlsx", sheet = "2019_Calf_info")%>%
  select(-percent)%>%
  rename(tag_num = 1, cow_weight = 2, calf_wean_weight = 3)

d_2019_2<-read_xlsx("calf_info.xlsx", sheet = "2019_Calf_info_2")%>%
  select(`Tag #`, DOB, Sex, BW, Dam, Sire...8)%>%
  rename(tag_num = 1, dob = 2, sex = 3, calf_birth_weight =4, dam = 5, sire = 6)
```

```
## New names:
## * Sire -> Sire...7
## * Sire -> Sire...8
```

```
d_2019<-d_2019_1%>%
  left_join(d_2019_2)%>%
  mutate(year = 2019,
         dob = as.Date(dob),
         wean_date = as.Date("2019-10-10"))%>%
  select(tag_num, sex, dob, calf_birth_weight, calf_wean_weight, wean_date, cow_weight, year, dam, sire)
```

```
## Joining, by = "tag_num"
```

```
d_2018_1<-read_xlsx("calf_info.xlsx", sheet = "2018_Calf_info")%>%
  select(`Tag#`, Birthdate, Sex, Birth, dam, sire)%>%
  rename(tag_num = 1, dob = 2, sex = 3, calf_birth_weight = 4)

d_2018_2<-read_xlsx("calf_info.xlsx", sheet = "2018_Calf_info2")%>%
  rename(tag_num = 1, calf_wean_weight = 3, cow_weight = 4)%>%
  select(-2, -5)

d_2018<-d_2018_1%>%
  left_join(d_2018_2)%>%
  mutate(year = 2018,
         dob = as.Date(paste0("2018-", month(dob), "-", day(dob))),
         wean_date = as.Date("2018-10-23"))%>%
  select(tag_num, sex, dob, calf_birth_weight, calf_wean_weight, wean_date, cow_weight, year, dam, sire)
```

```
## Joining, by = "tag_num"
```

```
d_2017_1<-read_xlsx("calf_info.xlsx", sheet = "2017_Calf_info")%>%
  rename(tag_num = 1, dob = 2, sex = 3, calf_birth_weight = 4, calf_wean_weight = 5)%>%
  mutate(dob = as.Date(dob))

d_2017_2<-read_xlsx("calf_info.xlsx", sheet = "2017_Calf_info2")%>%
  rename(tag_num = 1, cow_weight = `Cow weight`)%>%
  select(tag_num, cow_weight)%>%
  mutate(cow_weight = ifelse(cow_weight %in% c("?", "didn't get lbs"), NA, cow_weight),
         cow_weight = as.numeric(cow_weight))%>%
  filter(!is.na(tag_num))

d_2017<-d_2017_1%>%
  left_join(d_2017_2)%>%
  mutate(year = 2017,
         wean_date = as.Date("2017-10-20"))%>%
  select(tag_num, sex, dob, calf_birth_weight, calf_wean_weight, wean_date, cow_weight, year, dam, sire)
```

```
## Joining, by = "tag_num"
```

My original thought was to write a function that would using `lapply` and `excel_sheets`. But the each sheet needed custom cleaning so I ended up cleaning the sheets one by one.

Each years data is stored in two sheets. I read each years sheets one by one and did some combination of the following steps:

- Read in the data
- renamed column headers with `dplyr::renam`
- selected only the useful columns with `dplyr::select`
- joined the two datasheets from each year together with `dplyr::left_join`
- used `dplyr::mutate` to add a year and wean data (my sister gave me the wean dates in an email)

Now to add them all together and calculate a few helpful ratios and metrics. Some of these were just for my sister and won't get used.

```
data<-bind_rows(d_2019, d_2018, d_2017)%>%
  mutate(wean_age = as.numeric(wean_date-dob),
         calf_weight_gained = calf_wean_weight - calf_birth_weight,
         calf_weight_gained_per_day = calf_weight_gained/wean_age,
         weight_gained_as_perc_of_cow = calf_weight_gained/cow_weight,
         weight_gained_as_perc_of_cow_per_day = calf_weight_gained/cow_weight/wean_age)%>%
  mutate(id = paste0(tag_num, "_", year))%>%
  filter(id !="5715_2017")

rm(d_2017, d_2017_1, d_2017_2, d_2018, d_2018_1, d_2018_2, d_2019, d_2019_1, d_2019_2)

head(data)
```

```
## # A tibble: 6 x 16
##   tag_num sex   dob        calf_birth_weig~ calf_wean_weight wean_date
##   <chr>   <chr> <date>                <dbl>            <dbl> <date>
## 1 405     <NA>  NA                       NA              357 2019-10-10
## 2 502     H     2019-04-08               61              535 2019-10-10
## 3 503     H     2019-04-25               69              484 2019-10-10
## 4 504     S     2019-04-29               84              574 2019-10-10
## 5 514     S     2019-03-15               75              630 2019-10-10
## 6 515     H     2019-03-29               66              532 2019-10-10
## # ... with 10 more variables: cow_weight <dbl>, year <dbl>, dam <chr>,
## #   sire <chr>, wean_age <dbl>, calf_weight_gained <dbl>,
## #   calf_weight_gained_per_day <dbl>, weight_gained_as_perc_of_cow <dbl>,
## #   weight_gained_as_perc_of_cow_per_day <dbl>, id <chr>
```

And boom we have all of the data in one file. There was a problem with one of the cows in the spreadsheet so I had to remove it in the last step (cow 5715 from 2017).

Let's look at the dataset:

Variable	Description
tag_num	Calf Tag Number
sex	Calf sex
dob	Calf date of birth
calf_birth_weight	Is the first weight of the calf
calf_wean_weight	The weight of the calf when they are weaned
wean_date	Date the calf was weaned
cow_weight	Weight of the cow at weaning (I'm not totally sure when this measurement is taken)
year	The year of the data
dam	The tag number of the dam (or mother cow)
sire	The tag number of the sire (or father cow)

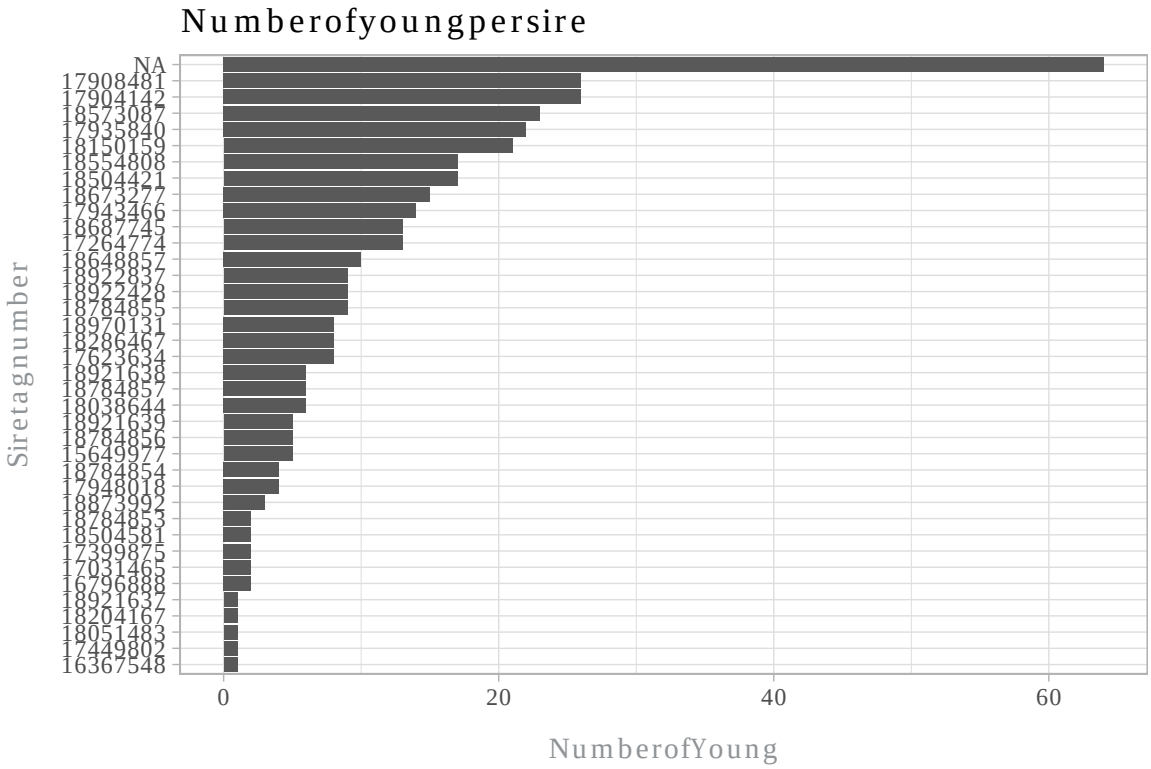
Variable	Description
wean_age	wean_date - dob (in days)
calf_weight_gained	calf_wean_weight - calf_birth_weight (ibs)
calf_weight_gained_per_day	calf_weight_gained/wean_age (ibs)
weight_gained_as_perc_of_cow	calf_weight_gained/cow_weight (ibs)
weight_gained_as_perc_of_cow_per_day	calf_weight_gained/cow_weight/wean_age
id	Observation unique id (tag_num combined with year)

Looking at the data

```
library(extrafont)
```

```
## Registering fonts with R
```

```
##font_import() -- You'll have to run this before you can use all of the fonts on your machine.
theme_set(theme_light()+
            theme(
              plot.margin = margin(20,20,20,20, unit = "pt"),
              text = element_text(family = "Noto Sans"),
              plot.title = element_text(face = "bold"),
              axis.title = element_text(color = "#909497"),
              axis.title.x = element_text(margin = margin(10,0,0,0, unit = "pt")),
              axis.title.y = element_text(margin = margin(0,10,0,0, unit = "pt")),
              legend.title = element_text(face = "bold")
            ))

data%>%
  count(sire, sort = T)%>%
  mutate(sire = fct_reorder(sire, n))%>%
  ggplot(aes(sire, n))+
  geom_col()+
  coord_flip()+
  labs(title = "Number of young per sire",
       x = "Sire tag number",
       y = "Number of Young")
```



Sires have up to 26 young. Many have had less than 3, however. There are also a lot of NAs.

Next lets look at dams.

```
data%>%
  count(dam, sort = T)%>%
  head()
```

```
## # A tibble: 6 x 2
##   dam          n
##   <chr>    <int>
## 1 <NA>        36
## 2 15163107     3
## 3 15440650     3
## 4 15453255     3
## 5 15817878     3
## 6 16196178     3
```

```
data%>%
  count(dam, sort = T)%>%
  count(n)%>%
  mutate(n = ifelse(n==36, NA, n))
```

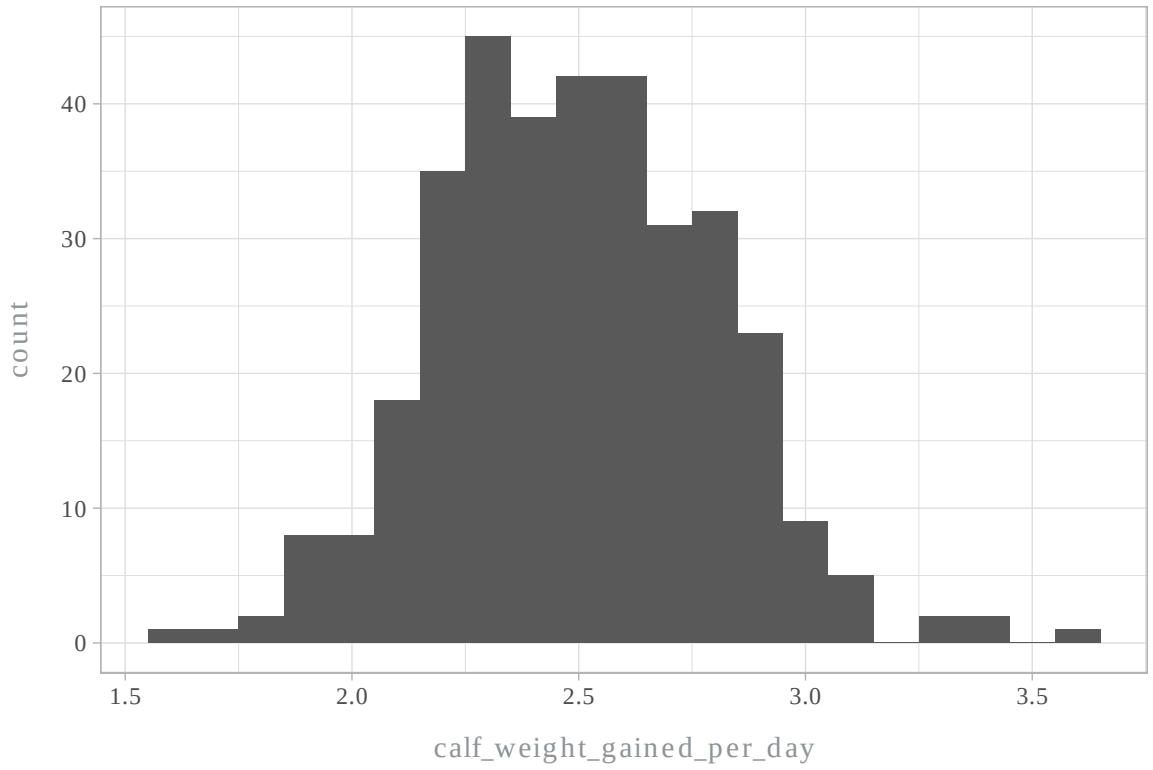
```
## # A tibble: 4 x 2
##       n    nn
##   <int> <int>
```

```
## 1     1    62
## 2     2    49
## 3     3    65
## 4    NA     1
```

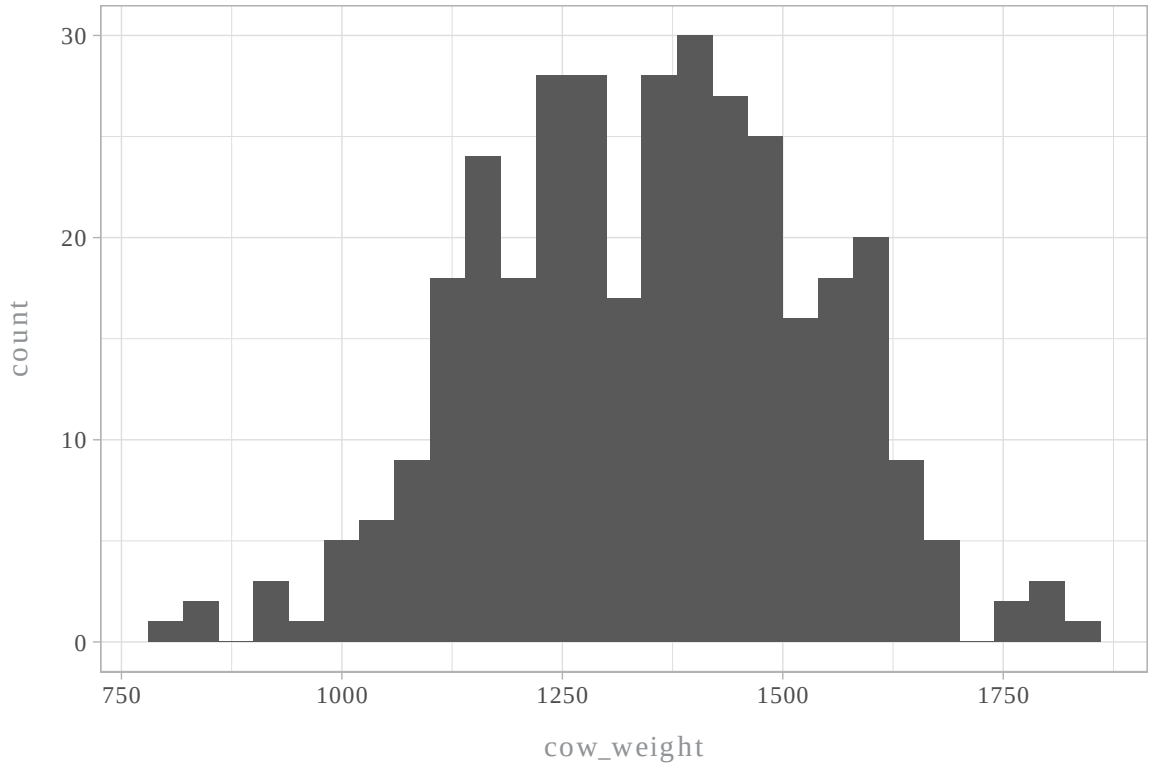
This one I am not going to plot because there are too many. Here we have almost 180 dams. Dams have a max of three calfs, which makes sense given that the dataset is from 2017 to 2019.

Let's also look at calf weight gained per day by cow weight.

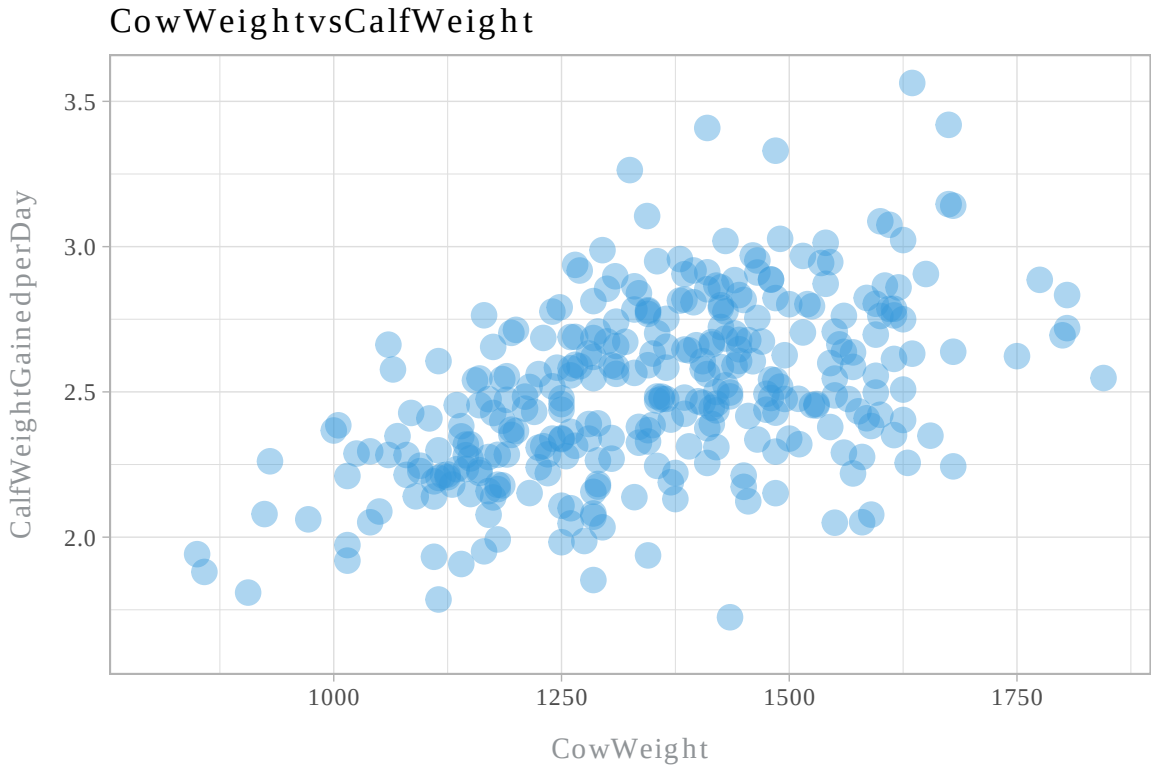
```
data%>%
  ggplot(aes(calf_weight_gained_per_day))+
  geom_histogram(binwidth = 0.1)
```




```
data%>%
  ggplot(aes(cow_weight))+
  geom_histogram(binwidth = 40)
```



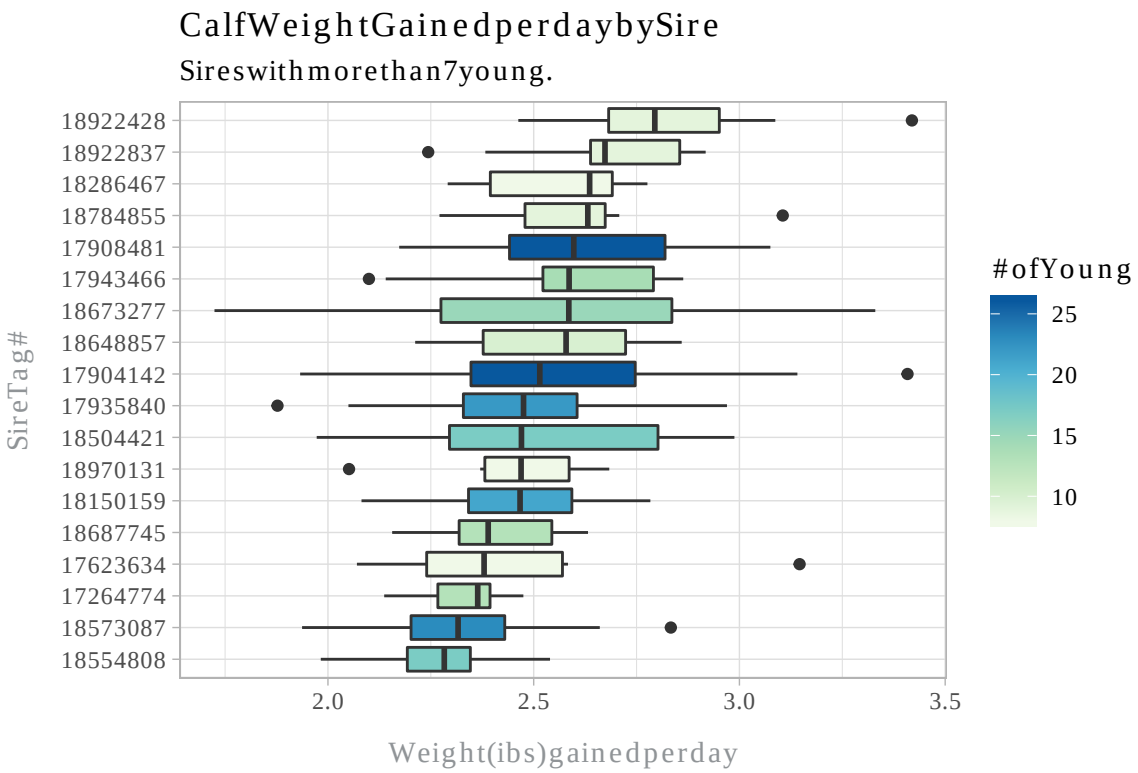

```
data%>%
  ggplot(aes(cow_weight, calf_weight_gained_per_day))+
  geom_point(color = "#3498DB", size = 4, alpha = 0.4)+
  labs(title = "Cow Weight vs Calf Weight",
       y = "Calf Weight Gained per Day",
       x = "Cow Weight")
```



It looks like both cow weight and calf weight are fairly normal and that cow weight and calf weight gained per day are fairly well correlated.

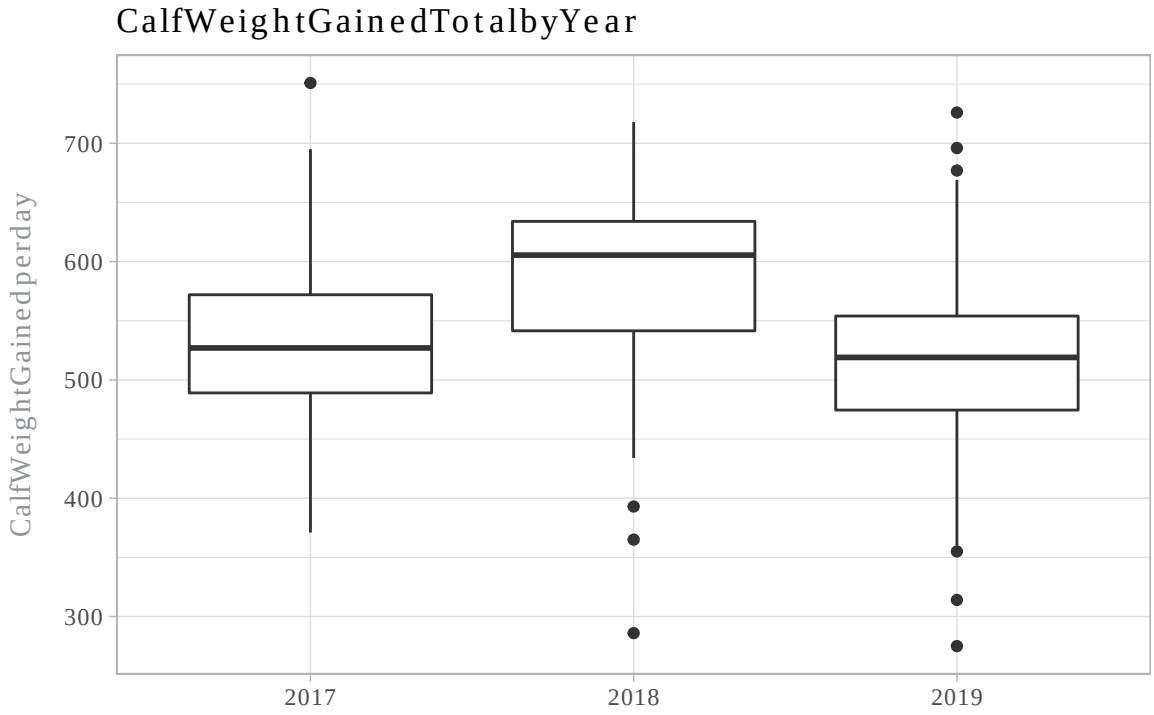
Now let's look at the weight gained per day of calfs produced by each sire that has had more than 7 young.

```
data%>%
  mutate(sire = fct_reorder(sire, calf_weight_gained_per_day, .fun = median, na.rm = T))%>%
  group_by(sire)%>%
  mutate(sire_count = n())%>%
  filter(n()>7)%>%
  ungroup()%>%
  filter(!is.na(sire))%>%
  ggplot(aes(sire, calf_weight_gained_per_day, fill = sire_count))+
  geom_boxplot()+
  coord_flip()+
  scale_fill_distiller(palette = "GnBu", direction = 1)+
  labs(title = "Calf Weight Gained per day by Sire",
       subtitle = "Sires with more than 7 young.",
       x = "Sire Tag #",
       y = "Weight (ibs) gained per day",
       fill = "# of Young")
```



I also wanted to look at how much cows gained per year to see if my sister was doing a good job of adding fat to her cows.

```
data%>%
  ggplot(aes(as.factor(year), calf_weight_gained, group = year))+
  geom_boxplot()+
  labs(title = "Calf Weight Gained Total by Year",
       x = " ",
       y = " Calf Weight Gained per day")
```



It looks like between 2018 cows were gaining more weight per day than either 2017 or 2017.

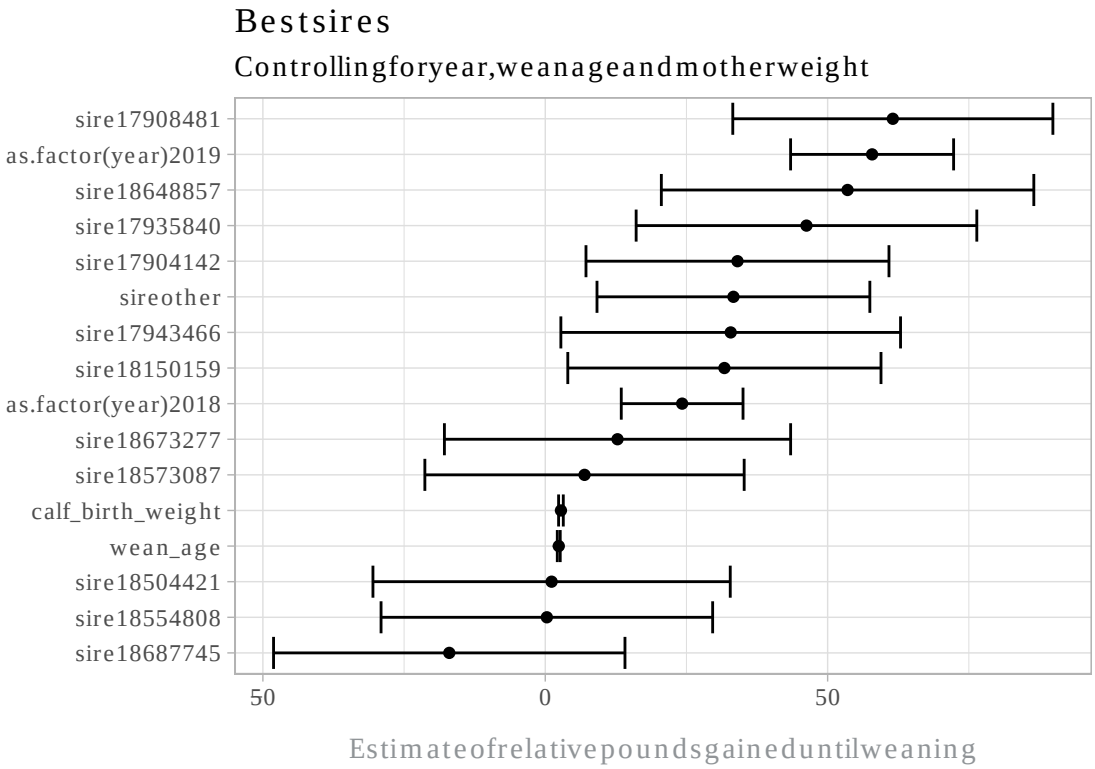
Linear Model

My sister wants to know which dams and sires produce calfs that weigh the most. The dams have only had 3 calfs since she took over the ranch. That's not really enough. But the Sires have had up to 26 young. So it seems that we could predict with a linear regression, calf weight gained per day using year, sex, wean age, birth weight, cow weight and sire. I'm going to limit the sires to only those that have had more than 10 young.

```
lm_data<-data%>%
  group_by(sire)%>%
  mutate(count = n())%>%
  ungroup()%>%
  mutate(sire = ifelse(count<10, "other", sire))%>%
  mutate(sire = replace_na(sire, "other"))%>%
  group_by(dam)%>%
  mutate(count = n())%>%
  ungroup()%>%
  mutate(dam = ifelse(count<3 | is.na(dam), "other", dam))%>%
  ungroup()

sm<-lm(calf_wean_weight ~ sire+wean_age+as.factor(year)+calf_birth_weight, data = lm_data)

summary(sm)
```

```
##
## Call:
## lm(formula = calf_wean_weight ~ sire + wean_age + as.factor(year) +
##     calf_birth_weight, data = lm_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -151.459  -37.287    1.683   34.460  148.840
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -161.2782    55.2887  -2.917  0.00378 **
## sire17904142          34.0345    20.8882   1.629  0.10419
## sire17908481          61.5428    22.0706   2.788  0.00560 **
## sire17935840          46.2527    23.4868   1.969  0.04976 *
## sire17943466          32.8356    23.4188   1.402  0.16183
## sire18150159          31.7185    21.5941   1.469  0.14283
## sire18504421           1.1206    24.6349   0.045  0.96375
## sire18554808           0.2806    22.8625   0.012  0.99022
## sire18573087           6.9507    22.0146   0.316  0.75241
## sire18648857          53.5292    25.6813   2.084  0.03790 *
## sire18673277          12.7932    23.8708   0.536  0.59237
## sire18687745         -16.9985    24.2272  -0.702  0.48341
## sireother             33.3168    18.8127   1.771  0.07749 .
## wean_age               2.3889     0.2028  11.779  < 2e-16 ***
## as.factor(year)2018   24.2374     8.3948   2.887  0.00414 **
## as.factor(year)2019   57.8722    11.2402   5.149 4.52e-07 ***
## calf_birth_weight      2.7756     0.3251   8.537 5.18e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 56.22 on 329 degrees of freedom
##   (45 observations deleted due to missingness)
## Multiple R-squared:  0.5732, Adjusted R-squared:  0.5524
## F-statistic: 27.62 on 16 and 329 DF,  p-value: < 2.2e-16
```

```
library(broom)

tidy(sm, conf.int = T, conf.level = .80)%>%
  filter(term != "(Intercept)")%>%
  mutate(term = fct_reorder(term, estimate))%>%
  ggplot(aes(term, estimate))+
  geom_point()+
  coord_flip()+
  geom_errorbar(aes(ymin = conf.low, ymax = conf.high))+
  labs(title = "Best sires",
       subtitle = "Controlling for year, wean age and mother weight",
       y = "Estimate of relative pounds gained until weaning",
       x = "")
```



Controlling for wean_age, year and calf birth weight you can see wich cows produce the heaviest young. Most of the sires are not statistically significant however. I would say that these results provide some evidence that at least some cows produce larger cows than others.