Assginment03

Zhen Liu

2022-10-14

1. Exploratory data analysis

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──
## ✔ ggplot2 3.3.6      ✔ purrr   0.3.4
## ✔ tibble  3.1.8      ✔ dplyr   1.0.10
## ✔ tidyr   1.2.1      ✔ stringr 1.4.1
## ✔ readr   2.1.3      ✔ forcats 0.5.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

```
library(Stat2Data)
data("Hawks")
```

```
head(Hawks)
```

```
##   Month Day Year CaptureTime ReleaseTime  BandNumber Species Age Sex Wing
## 1     9  19 1992       13:30                877-76317      RT   I       385
## 2     9  22 1992       10:30                877-76318      RT   I       376
## 3     9  23 1992       12:45                877-76319      RT   I       381
## 4     9  23 1992       10:50                745-49508      CH   I   F   265
## 5     9  27 1992       11:15               1253-98801      SS   I   F   205
## 6     9  28 1992       11:25              1207-55910       RT   I       412
##   Weight Culmen Hallux Tail StandardTail Tarsus WingPitFat KeelFat Crop
## 1    920   25.7   30.1  219           NA     NA         NA      NA   NA
## 2    930     NA     NA  221           NA     NA         NA      NA   NA
## 3    990   26.7   31.3  235           NA     NA         NA      NA   NA
## 4    470   18.7   23.5  220           NA     NA         NA      NA   NA
## 5    170   12.5   14.3  157           NA     NA         NA      NA   NA
## 6   1090   28.5   32.2  230           NA     NA         NA      NA   NA
```

1.1 Location estimators

Q1

```
HawksTail<-Hawks[['Tail']]

HawksTail
```

```
##   [1] 219 221 235 220 157 230 212 243 210 238 222 217 213 238 243 232 238 202
##  [19] 227 227 222 237 238 213 211 130 190 245 164 246 207 209 200 215 219 198
##  [37] 207 204 205 144 136 191 230 227 208 231 222 225 225 233 214 233 158 245
##  [55] 210 230 192 229 235 221 210 212 225 215 122 235 232 133 154 209 212 209
##  [73] 250 235 222 236 210 239 228 220 233 236 155 152 135 186 216 233 248 221
##  [91] 227 223 219 225 238 235 222 155 160 137 127 238 150 150 229 227 129 220
## [109] 245 223 224 133 210 234 219 216 230 223 220 241 136 137 223 238 126 235
## [127] 240 137 160 140 220 218 234 232 221 220 232 214 225 238 133 136 151 131
## [145] 238 229 202 226 220 215 122 134 215 208 211 220 204 229 205 155 150 244
## [163] 225 239 222 209 164 159 211 160 157 160 216 250 260 164 217 228 218 220
## [181] 227 211 222 225 221 231 235 182 235 200 216 223 210 229 160 125 226 154
## [199] 207 238 151 158 197 250 215 214 230 213 214 224 153 131 145 165 168 156
## [217] 215 164 155 239 185 214 232 205 243 255 159 156 276 210 145 231 216 210
## [235] 210 220 260 200 213 230 220 220 235 215 215 221 225 238 221 212 215 235
## [253] 132 229 251 215 225 210 267 220 238 248 198 241 212 221 233 138 230 223
## [271] 226 208 231 217 230 153 186 225 226 214 242 220 250 178 220 240 215 160
## [289] 230 220 234 226 218 209 228 235 223 245 202 235 158 133 221 125 158 159
## [307] 132 136 216 225 137 155 155 138 130 154 216 200 158 132 215 226 216 221
## [325] 196 221 218 157 135 221 210 220 220 221 242 220 210 221 236 162 150 235
## [343] 231 227 238 225 249 213 240 231 227 155 227 223 228 242 149 184 247 158
## [361] 267 257 148 221 214 229 213 138 213 158 193 201 216 165 160 244 230 227
## [379] 139 233 240 237 144 210 225 125 246 225 230 220 230 233 196 135 170 232
## [397] 234 239 230 227 213 207 214 146 217 216 214 216 222 288 238 199 161 235
## [415] 208 137 143 219 128 161 236 131 225 190 150 223 225 241 136 210 160 135
## [433] 223 219 229 211 204 204 146 214 236 185 195 247 215 227 155 159 132 225
## [451] 130 220 215 161 226 227 236 169 212 210 164 226 242 153 221 220 219 225
## [469] 221 215 223 219 235 216 151 130 154 223 221 220 186 217 209 215 214 169
## [487] 207 153 210 224 200 207 137 212 132 226 232 228 205 155 228 237 135 230
## [505] 219 223 210 222 124 201 206 234 149 204 204 154 224 150 147 130 180 229
## [523] 153 187 222 214 132 219 230 219 230 217 179 225 123 229 215 247 222 220
## [541] 234 192 248 221 132 162 218 225 215 225 244 232 238 231 218 223 152 197
## [559] 195 206 213 127 122 222 124 227 235 158 150 133 228 219 231 127 127 209
## [577] 223 237 210 133 215 217 135 207 211 225 131 203 223 218 222 235 208 216
## [595] 157 217 219 235 185 212 211 201 200 137 217 223 199 217 207 222 163 217
## [613] 200 220 216 216 222 136 142 213 154 226 156 160 149 131 137 126 210 165
## [631] 235 222 129 192 156 135 129 131 160 195 223 192 198 152 223 132 216 207
## [649] 218 215 224 242 157 119 130 124 130 237 140 159 138 159 136 221 243 222
## [667] 208 132 245 131 206 220 214 217 162 210 163 196 206 218 218 210 224 154
## [685] 161 209 157 158 156 133 218 213 165 235 206 206 153 135 207 230 224 216
## [703] 218 215 226 242 212 193 201 129 216 132 131 234 220 225 214 230 220 200
## [721] 209 156 225 139 159 226 154 230 131 156 233 235 226 145 156 130 130 147
## [739] 141 231 205 192 225 205 226 156 153 131 155 230 165 238 132 233 135 134
## [757] 181 217 227 230 136 217 154 216 215 199 210 217 212 223 215 196 218 221
## [775] 215 163 218 204 151 218 155 216 156 220 230 213 220 213 215 214 136 215
## [793] 217 158 219 215 211 234 161 208 153 152 163 226 125 134 162 185 212 123
## [811] 228 221 131 143 129 223 232 125 214 150 183 158 230 225 141 183 224 162
## [829] 131 188 132 215 237 133 156 163 134 253 211 160 158 187 196 217 227 220
## [847] 227 238 122 201 185 137 220 212 197 230 147 233 135 140 218 222 233 159
## [865] 136 222 203 218 242 129 205 236 121 134 233 211 186 217 218 241 218 208
## [883] 212 152 203 218 153 196 184 156 217 212 237 206 158 157 157 201 158 224
## [901] 199 219 217 224 150 211 207 222
```

```
class(HawksTail)
```

```
## [1] "integer"
```

```
mean(HawksTail)
```

```
## [1] 198.8315
```

```
median(HawksTail)
```

```
## [1] 214
```

1.2 Combining location estimators with the summarise function

Q1

```
Hawks%>%
  summarise(Wing_mean= mean(Wing,na.rm = TRUE),Wing_t_mean = mean(Wing,na.rm = TRUE,t
rim = 0.5), Wing_med = median(Wing,na.rm = TRUE),Weight_mean= mean(Weight,na.rm = TRU
E),Weight_t_mean = mean(Weight,na.rm = TRUE,trim = 0.5), Weight_med = median(Weight,n
a.rm = TRUE))
```

```
##   Wing_mean Wing_t_mean Wing_med Weight_mean Weight_t_mean Weight_med
## 1  315.6375         370      370    772.0802           970        970
```

Q2

```
Hawks%>%
  group_by(Species)%>%
  summarise(Wing_mean= mean(Wing,na.rm = TRUE),Wing_t_mean = mean(Wing,na.rm = TRUE,t
rim = 0.5), Wing_med = median(Wing,na.rm = TRUE),Weight_mean= mean(Weight,na.rm = TRU
E),Weight_t_mean = mean(Weight,na.rm = TRUE,trim = 0.5), Weight_med = median(Weight,n
a.rm = TRUE))
```

```
## # A tibble: 3 × 7
##   Species Wing_mean Wing_t_mean Wing_med Weight_mean Weight_t_mean Weight_med
##   <fct>       <dbl>       <dbl>    <dbl>       <dbl>         <dbl>      <dbl>
## 1 CH           244.         240      240        420.          378.       378.
## 2 RT           383.         384      384       1094.         1070       1070
## 3 SS           185.         191      191        148.          155        155
```

1.3 Location and dispersion estimators under linear transformations

Q1

```
a<- 2
b<-3

mean(HawksTail*a +b)
```

```
## [1] 400.663
```

```
mean(HawksTail)
```

```
## [1] 198.8315
```

The mean of HawksTail*a +b is almost double times as HawksTail

Q2

```
var(HawksTail*a +b)
```

```
## [1] 5424.147
```

```
var(HawksTail)
```

```
## [1] 1356.037
```

```
sd(HawksTail*a +b)
```

```
## [1] 73.64881
```

```
sd(HawksTail)
```

```
## [1] 36.8244
```

The variance of HawksTail*a +b is almost four times as HawksTail The standard deviation of HawksTail*a +b is almost two times as HawksTail

1.4 Robustness of location estimators

```
hal<-Hawks$Hallux # Extract the vector of hallux lengths
hal<-hal[!is.na(hal)] # Remove any nans
```

```
outlier_val<-100
num_outliers<-10

corrupted_hal<-c(hal,rep(outlier_val,times=num_outliers))

mean(hal)
```

```
## [1] 26.41086
```

```
mean(corrupted_hal)
```

```
## [1] 27.21776
```

```
num_outliers_vect <- seq(0,1000)
means_vect <- c()
for(num_outliers in num_outliers_vect){
  corrupted_hal <-c(hal,rep(outlier_val,times=num_outliers))
  means_vect <- c(means_vect, mean(corrupted_hal))
}
```

Q1

```
medians_vect<-c()
for(num_outliers in num_outliers_vect){
  corrupted_hal<-c(hal,rep(outlier_val,times=num_outliers))
  medians_vect<-c(medians_vect,median(corrupted_hal))
}
```

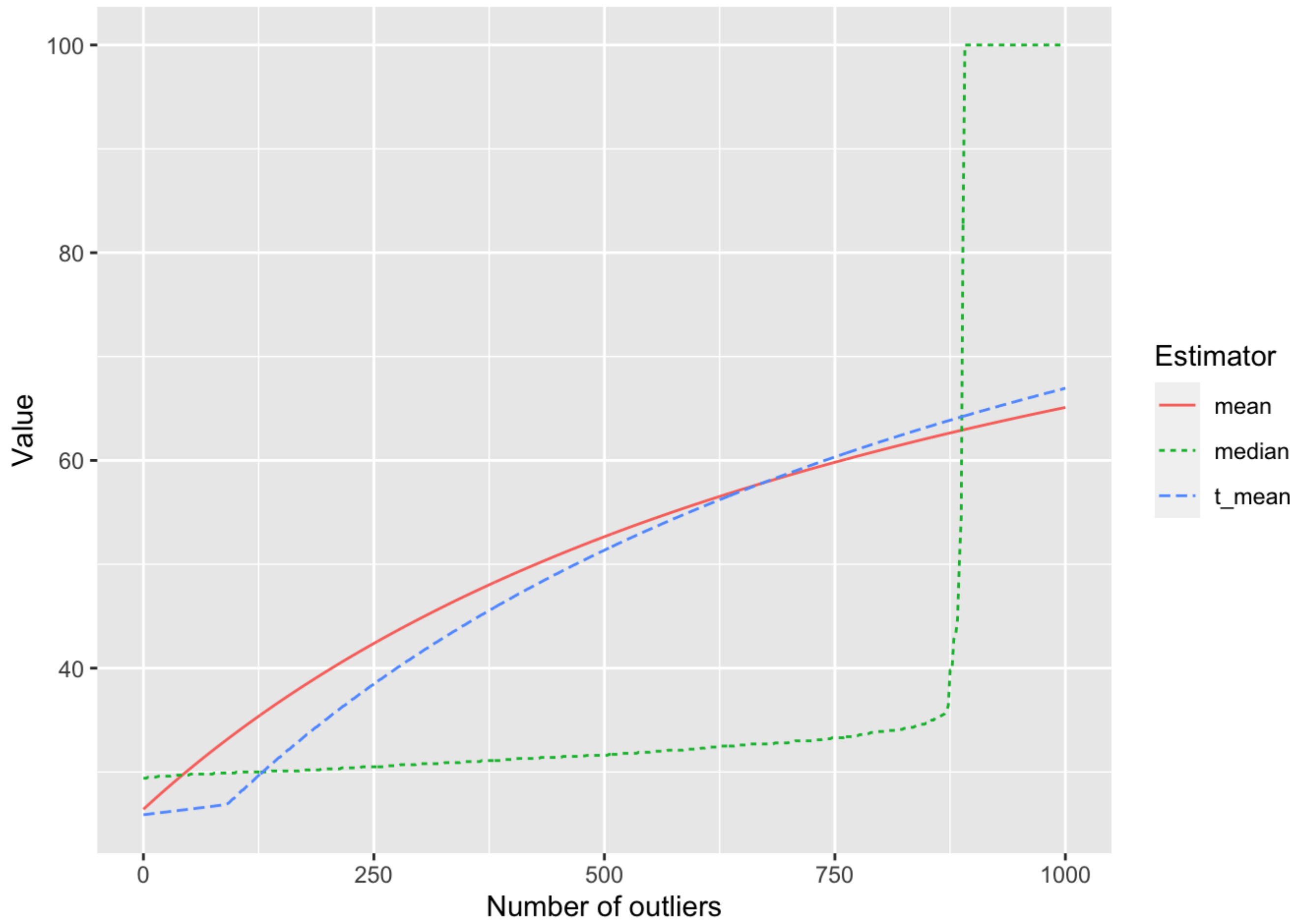
Q2

```
t_means_vect<-c()
for(num_outliers in num_outliers_vect){
  corrupted_hal<-c(hal,rep(outlier_val,times=num_outliers))
  t_means_vect<-c(t_means_vect,mean(corrupted_hal,trim = 0.1))
}
```

Q3

```
df_means_medians <-
  data.frame(num_outliers=num_outliers_vect,
             mean=means_vect, t_mean=t_means_vect,
             median=medians_vect)
```

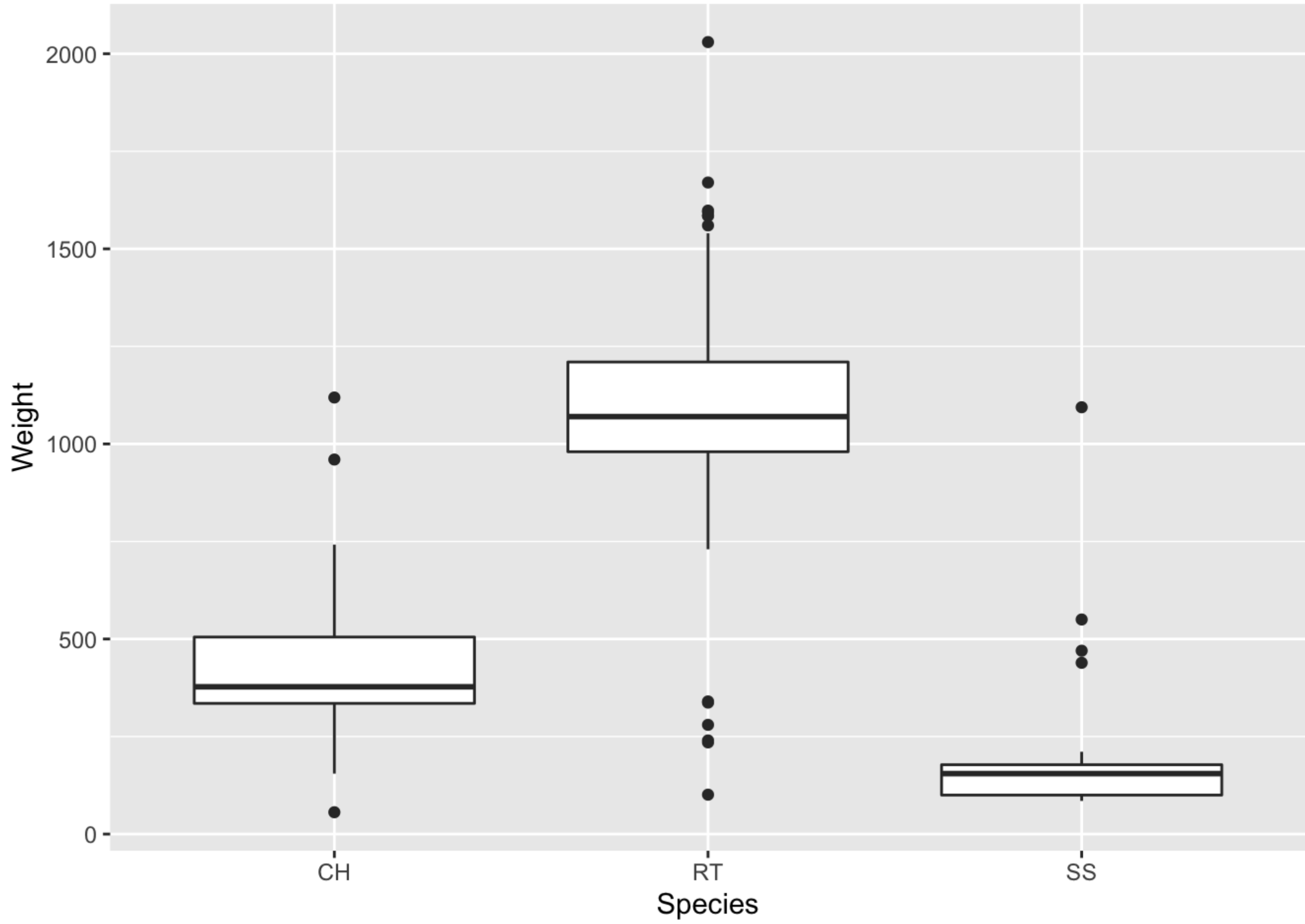
```
df_means_medians %>% pivot_longer(!num_outliers,
                                  names_to = "Estimator", values_to = "Value") %>%
  ggplot(aes(x=num_outliers,color=Estimator,
             linetype=Estimator,y=Value)) +
  geom_line()+xlab("Number of outliers")
```



1.5 Box plots and outliers

```
ggplot(Hawks,aes(x=Species,y=Weight)) +
  geom_boxplot()+xlab("Species") + ylab("Weight")
```

```
## Warning: Removed 10 rows containing non-finite values (stat_boxplot).
```



Q2 quantile annd boxplots

```
q =c(0.25,0.5,0.75)

Hawks%>%
  group_by(Species)%>%
  summarise(quantile025 = quantile(Weight, probs = q[1],na.rm=TRUE),
            quantile050 = quantile(Weight, probs = q[2],na.rm=TRUE),
            quantile075 = quantile(Weight, probs = q[3],na.rm=TRUE))
```

```
## # A tibble: 3 × 4
##   Species quantile025 quantile050 quantile075
##   <fct>         <dbl>       <dbl>       <dbl>
## 1 CH              335        378.         505
## 2 RT              980       1070         1210
## 3 SS              100        155          178.
```

Q3 outliers

```
num_outliers<- function(input_sample){
  quantile25<- quantile(input_sample,0.25, na.rm = TRUE)
  quantile75<- quantile(input_sample,0.75, na.rm = TRUE)
  iq_range<- quantile75 - quantile25
  outliers <- input_sample[(input_sample>quantile75+1.5*iq_range) | (input_sample<qua
ntile25-1.5*iq_range)]
  n<-length(which(!is.na(outliers)))
  return(n)
}

num_outliers(c(0,40,60,185))
```

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

Q4 Outliers by group

```
Hawks%>%
  group_by(Species)%>%
  summarise(num_outliers_weight=num_outliers(Weight))
```

```
## # A tibble: 3 × 2
##   Species num_outliers_weight
##   <fct>                 <int>
## 1 CH                        3
## 2 RT                       13
## 3 SS                        4
```

1.6 Covariance and correlation

Q1

```
cov(Hawks$Weight,Hawks$Wing,use = 'complete.obs')
```

```
## [1] 41174.39
```

```
cor(Hawks$Weight,Hawks$Wing,use = 'complete.obs')
```

```
## [1] 0.9348575
```

Q2

```
S_new_n<- function(X,Y,a,b,c,d){
  S <- cov(X,Y,use ='complete.obs' )
  x_new<- a*X+b
  y_new<- c*Y+d
  S_new <- abs(cov(x_new,y_new,use ='complete.obs'))

  return(S_new)
}
```

```
S_new_n(Hawks$Weight,Hawks$Wing,2.4,7.1,-1,3)
```

```
## [1] 98818.54
```

S_new is 2.4 times than S

2. Random experiments, events and sample spaces, and the set theory

2.1 Random experiments, events and sample sapces

Q1 Firstly, write down the definition of a random experiment, event and sample space.

A random experiment is a procedure (real or imagined) which: 1. has a well-defined set of possible outcomes; 2. could (at least in principle) be repeated arbitrarily many times.

An event is a set (i.e. a collection) of possible outcomes of an experiment

A sample space is the set of all possible outcomes of interest for a random experiment

Q2 Consider a random experiment of rolling a dice twice. Give an example of what is an event in this random experiment. Also, can you write down the sample space as a set? What is the total number of different events in this experiment? Is the empty set considered as an event?

Example event : get {6,6} Total number of different events : 36 you cannot get an empty set because you must get a number when you rolling a dice

2.2 Set theory

Q1 Set operations:

Let the sets A, B, C be defined by A := {1, 2, 3}, B := {2, 4, 6}, C := {4, 5, 6}.

1. What are the unions A∪B and A∪C?
2. What are the intersections A ∩ B and A∩C?
3. What are the complements Aand A?
4. Are A and B disjoint? Are A and C disjoint?
5. Are B and Adisjoint?
6. Write down a partition of {1,2,3,4,5,6} consisting of two sets. Also, write down another partition of {1,2,3,4,5,6} consisting of three sets

Answer:
1. unions A∪B : {1,2,3,4,6} A∪C :{1,2,3,4,5,6}
2. intersections A ∩ B : {2} A∩C empty set
3. complements A: {1,3} A: {1,2,3}
4. no / yes
5. yes
6. {1,2,3} {4,5,6}
{1,2} {3,4} {5,6}

Q2 Complements, subsets and De Morgan’s laws

Let Ω be a sample space. Recall that for an event A ⊆ Ω the complement Ac := Ω:= {w ∈ Ω : w /∈ A}. Take a pair of events A ⊆ Ω and B ⊆ Ω.

1. Can you give an expression for (Ac)c without using the notion of a complement?
2. What is Ωc?
3. (Subsets) Show that if A ⊆ B, then Bc ⊆ Ac.
4. (De Morgan's laws) Show that (A ∩ B)c = Ac ∪ Bc. Let's suppose we have a sequence of events A1,A2,··· ,AK ⊂ Ω. Can you write out an expression for (∩K k=1Ak)c?
5. (De Morgan's laws) Show that (A∪B)c = Ac ∩Bc.
6. Let's suppose we have a sequence of events A1,A2,··· ,AK ⊂ Ω. Can you write out an expression for (∪K k=1Ak)c?

Answer:
1. A
2. empty set
3. A: {1,2,3} B{1,2,3,4,5} Ω:{1,2,3,4,5,6,7,8,9} Ac :{4,5,6,7,8,9} Bc:{6,7,8,9}
Bc ⊆ Ac
4.

$$(\cap_{k=1}^{K} A_k)^c := \cup_{k=1}^{K} (A_k)^c$$

5.

$$A \cup B := \{1, 2, 3, 4, 5\}$$

$$(A \cup B)^c := \{6, 7, 8, 9\}$$

$$A^c := \{4, 5, 6, 7, 8, 9\} \; B^c := \{6, 7, 8, 9\} \; A^c \cap B^c := \{6, 7, 8, 9\}$$

6.

$$(\cup_{k=1}^{K} A_k)^c := \cap_{k=1}^{K} (A_k)^c$$

Q3 Cardinality and the set of all subsets:

$$E := \{A \in \Omega : A \in A_i \text{ for all i = 1,2,...,K } \}$$

Q4 Disjointness and partitions

1.

$$A_1 := \{1\}, A_2 =: \{2\}, A_3 =: \{3\}, A_4 =: \{4\}, \Omega := \{1, 2, 3, 4\}$$

2.

$$S_1 := \{1\}, S_2 := \{2\}, S_3 := \{3\}, S_4 := \{4\}$$

$$S_1, S_2, S_3, S_4 \text{ form a partition of} \{1, 2, 3, 4\}$$

Q5 Indicator function

1.

$$1_A{}^c(w) = \{ \quad \begin{array}{lll} 1_A & \text{if} & w! \in A \\ 1_A & \text{if} & w \in A \end{array}$$

2.

$$B = \Omega$$

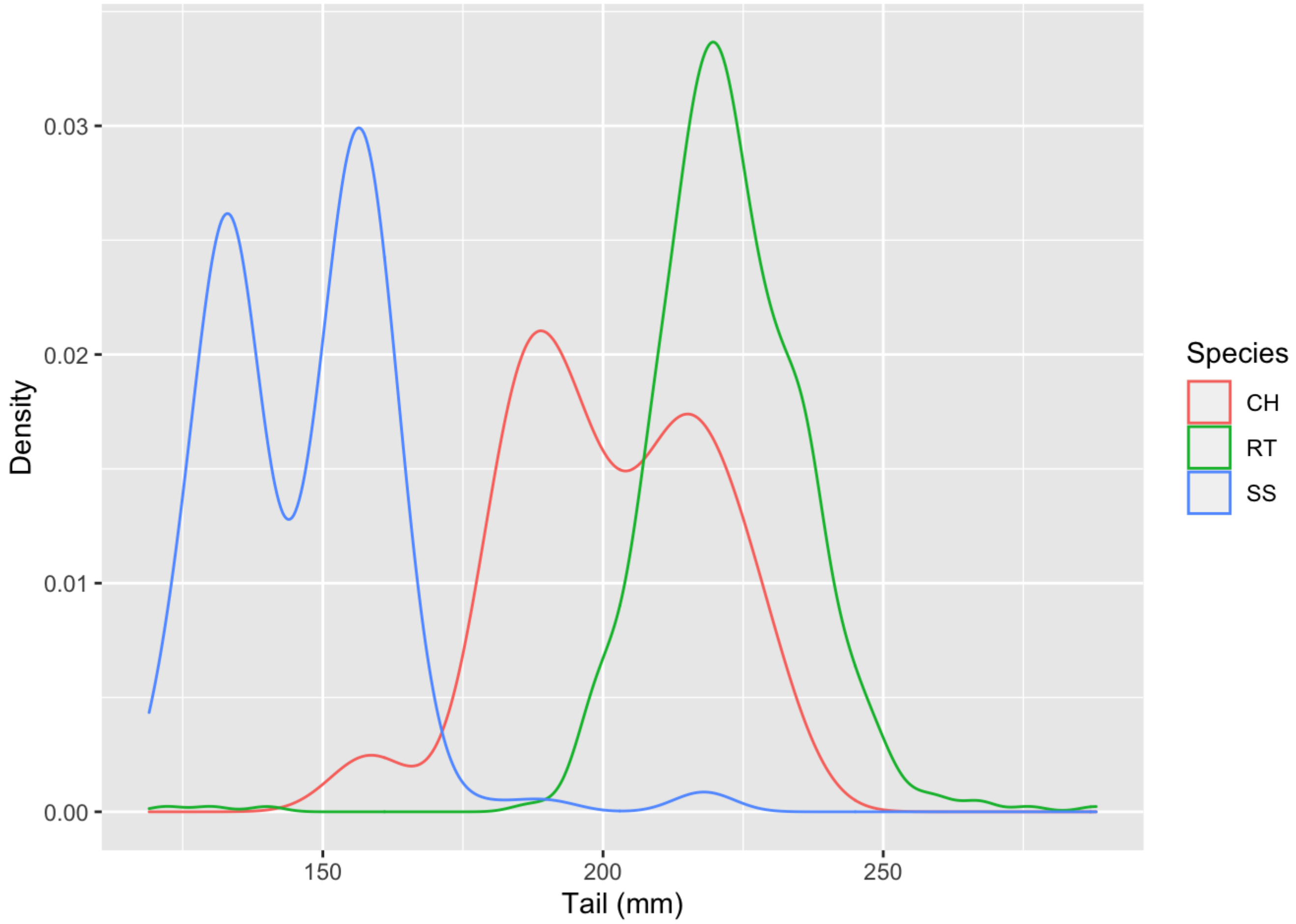
3.

Q6 Uncountable infinities

3.Visualisation

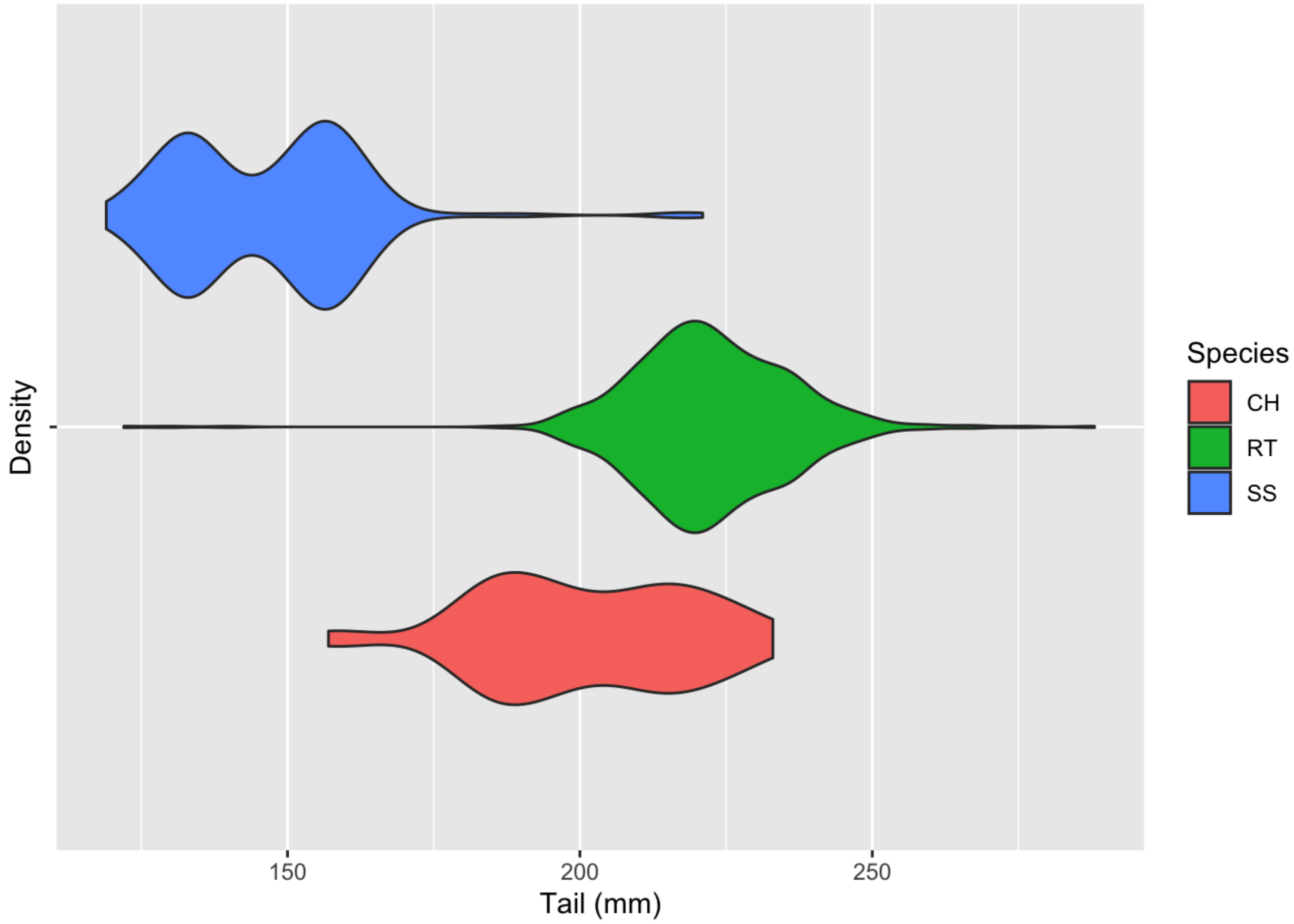
Q1 Density plot:

```
library(ggplot2)
ggplot(Hawks,aes(x=Tail,group=Species,color=Species)) + geom_density()+xlab("Tail (m
m)") + ylab("Density")
```



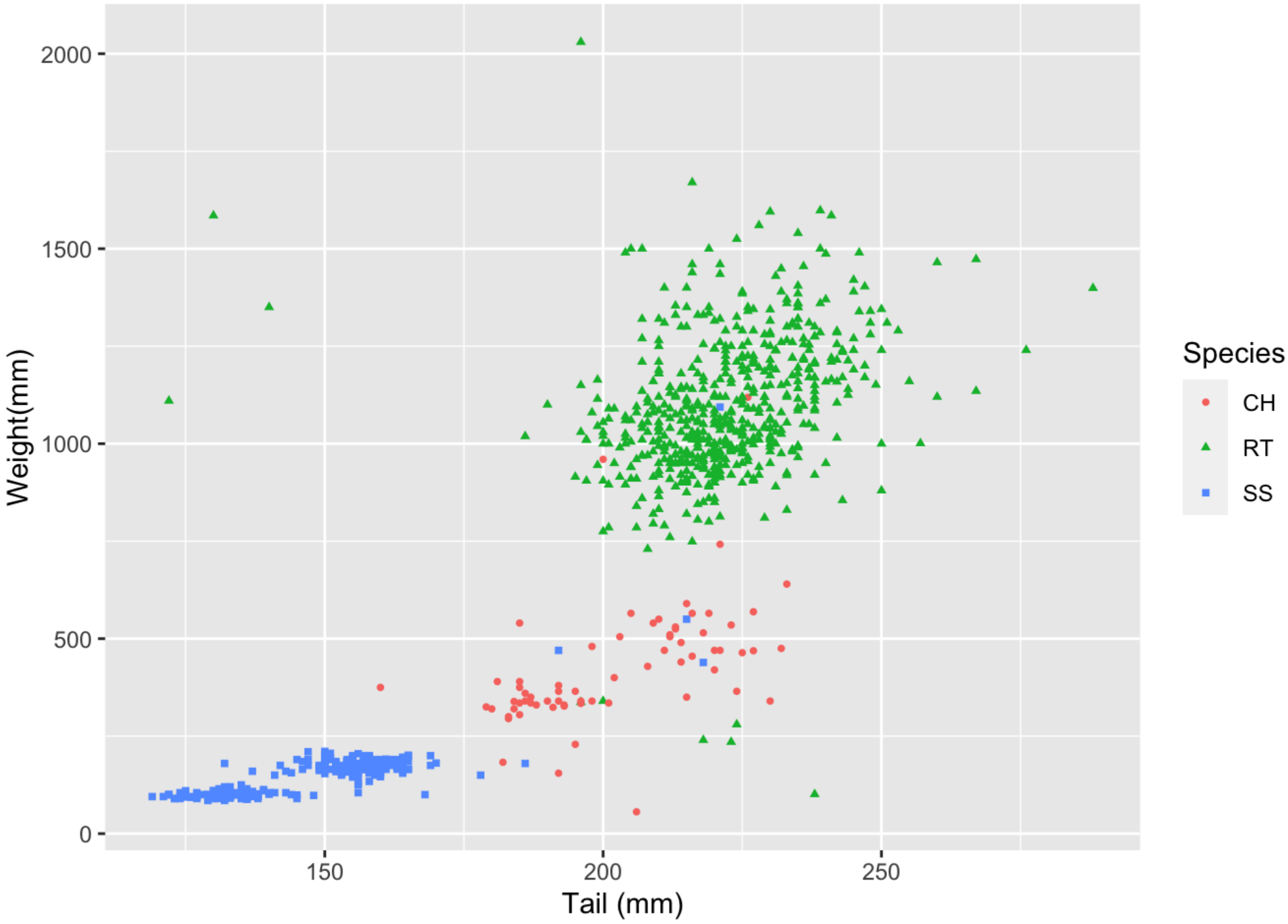
Q2 Violin plot:

```
ggplot(Hawks,aes(x=Tail,y="",fill=Species)) +
  geom_violin()+xlab("Tail (mm)") + ylab("Density")
```



Q3 Scatter plot:

```
ggplot(Hawks,aes(x=Tail,y=Weight,group=Species,color=Species,shape=Species)) + geom_p
oint(size =1,na.rm = TRUE)+xlab("Tail (mm)") + ylab("Weight(mm)")
```



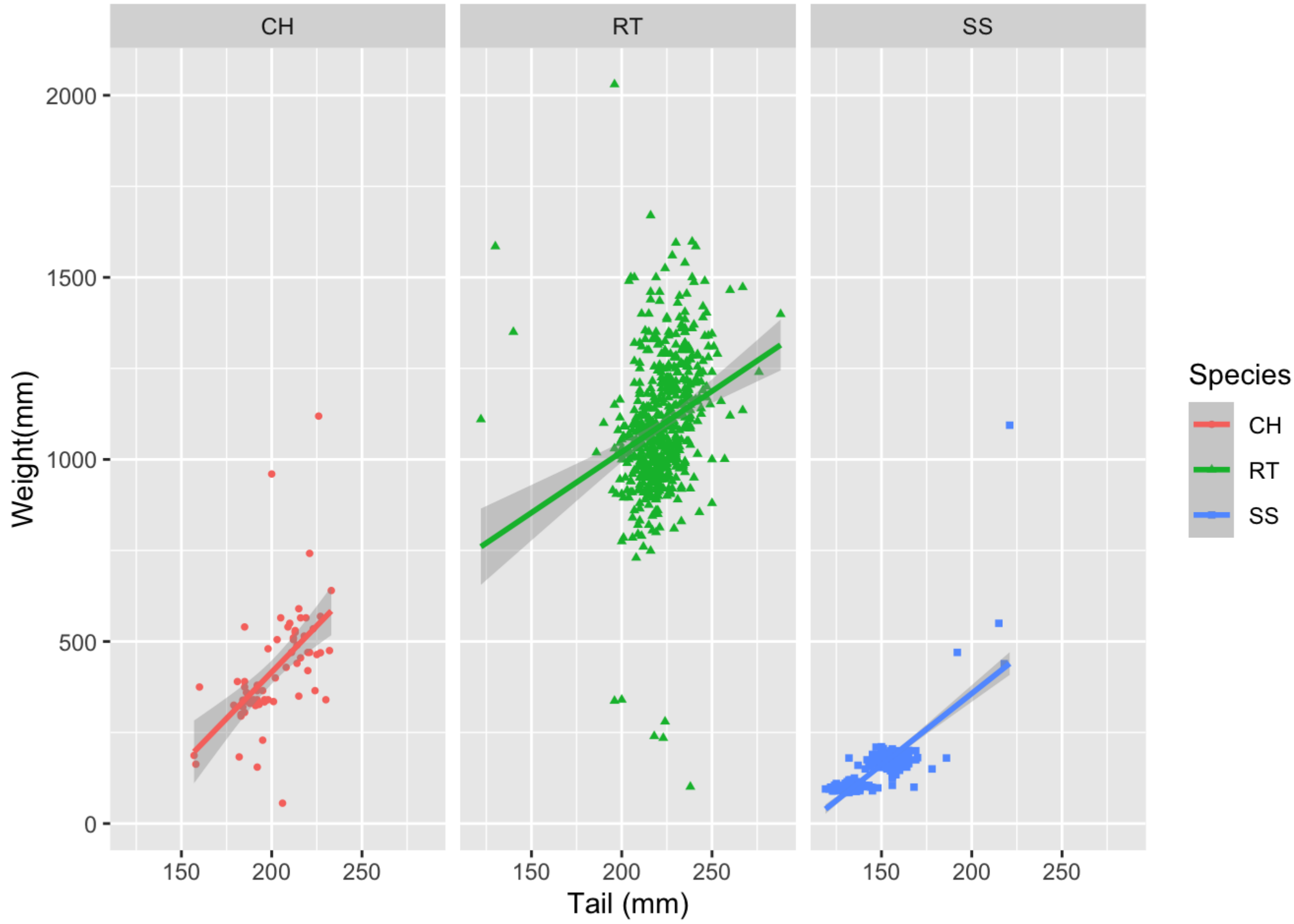
1.

5 types 2. key_points 3. Species

Q4

```
ggplot(Hawks,aes(x=Tail,y=Weight,group=Species,color=Species,shape=Species)) + geom_p
oint(size =1,na.rm = TRUE)+geom_smooth(method = "lm",na.rm = TRUE)+xlab("Tail (mm)")
 + ylab("Weight(mm)")+
  facet_wrap(~ Species)
```

```
## `geom_smooth()` using formula 'y ~ x'
```



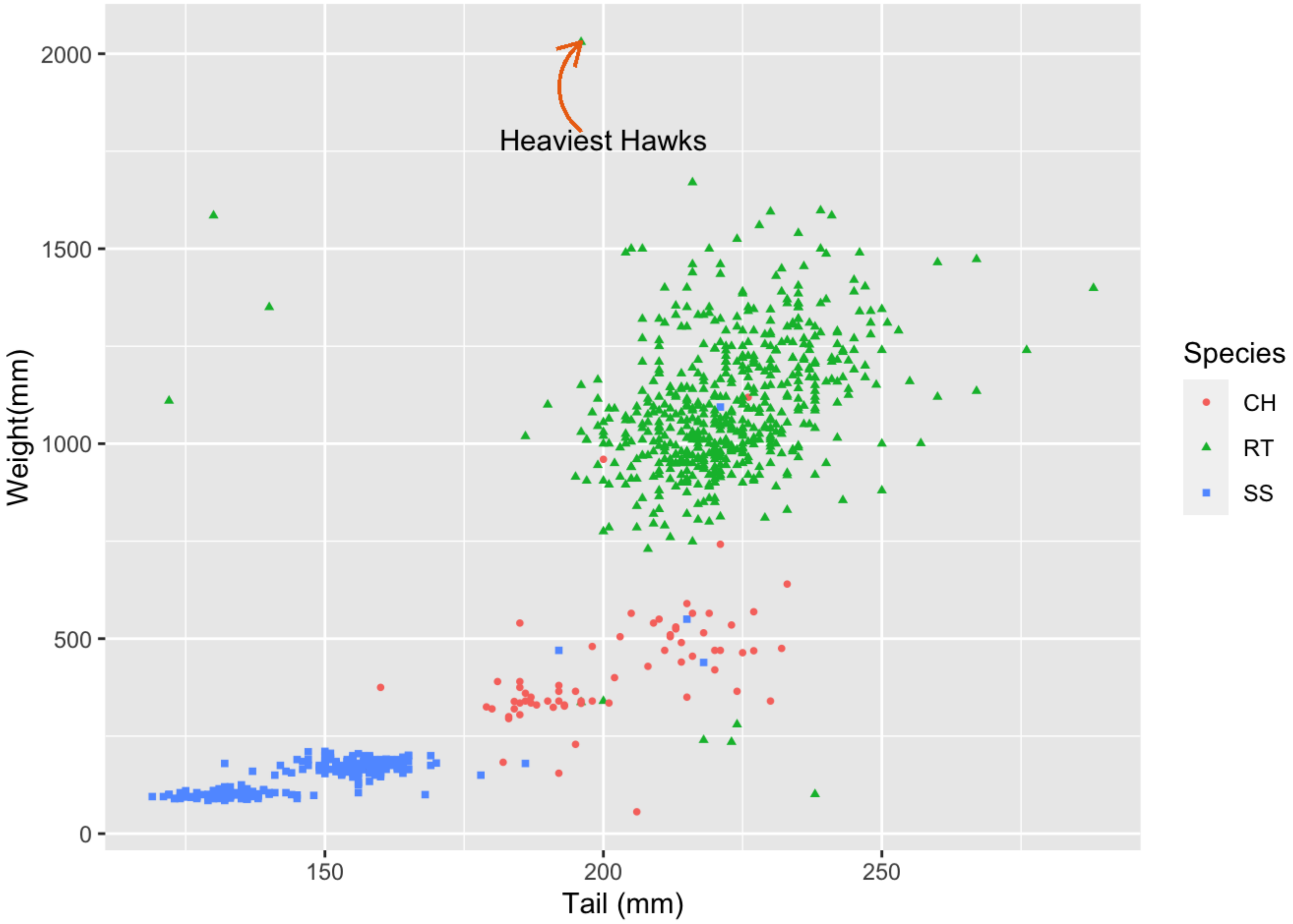
1. Species
2. positive correlation

Q5

```
library(tidyverse)
Hawks%>%
  select(Weight,Tail)%>%
  filter(Weight==max(Weight,na.rm = TRUE))
```

```
##   Weight Tail
## 1   2030  196
```

```
ggplot(Hawks,aes(x=Tail,y=Weight,group=Species,color=Species,shape=Species)) + geom_p
oint(size =1,na.rm = TRUE)+xlab("Tail (mm)") + ylab("Weight(mm)")+ geom_curve(
  aes(x = 196, y = 2030, xend =196, yend = 1800),
  arrow=arrow(ends ="first",length = unit(0.03,"npc"),type = "open"),
  colour = "#EC7014",
  size = 0.5,
  angle = 90 # Anything other than 90 or 0 can look unusual
)+annotate("text", x = 200, y = 1780, label = "Heaviest Hawks")
```


Heaviest Hawks
Weight(mm)
2000
1500
1000
500
0
Tail (mm)
150
200
250
Species
CH
RT
SS