EDA_abalone

Kim Namho

2024-06-04

본 문서에서는 다음과 같은 패키지를 활용합니다.

- skimr : Data Summary
- corrplot : Heat map of correlation matrix
- e1071 : Skewness
- GGally : ggpairs
- factoextra : PCA
- gridExtra : Arrange plots
- tidyverse : data handling
- flextable : flextable
- car : VIF

```
abalone <- read_csv("abalone1.data.txt", col_names = T)
glimpse(abalone)
```

```
## Rows: 4,177
## Columns: 9
## $ Sex            <chr> "M", "M", "F", "M", "I", "I", "F", "F", "M", "F", "F", …
## $ Length         <dbl> 0.455, 0.350, 0.530, 0.440, 0.330, 0.425, 0.530, 0.545,…
## $ Diameter       <dbl> 0.365, 0.265, 0.420, 0.365, 0.255, 0.300, 0.415, 0.425,…
## $ Height         <dbl> 0.095, 0.090, 0.135, 0.125, 0.080, 0.095, 0.150, 0.125,…
## $ Whole_weight   <dbl> 0.5140, 0.2255, 0.6770, 0.5160, 0.2050, 0.3515, 0.7775,…
## $ Shucked_weight <dbl> 0.2245, 0.0995, 0.2565, 0.2155, 0.0895, 0.1410, 0.2370,…
## $ Viscera_weight <dbl> 0.1010, 0.0485, 0.1415, 0.1140, 0.0395, 0.0775, 0.1415,…
## $ Shell_weight   <dbl> 0.150, 0.070, 0.210, 0.155, 0.055, 0.120, 0.330, 0.260,…
## $ Rings          <dbl> 15, 7, 9, 10, 7, 8, 20, 16, 9, 19, 14, 10, 11, 10, 10, …
```

각 변수 정보 및 기본 통계량

- Sex: 성별 (M: 수컷, F: 암컷, I: infant) type: categorical
- Length: 길이(mm) type: continuous
- Diameter: 지름(mm) type: continuous
- Height: 높이(mm) type: continuous
- Whole weight: 총 무게(grams) type: continuous

- Shucked weight: 껍데기를 벗긴 무게(grams) type: continuous
- Viscera weight: 내장 무게(grams) type: continuous
- Shell weight: 껍데기 무게(grams) type: continuous
- Rings: 수명 type: integer

```
# skim 함수를 사용하여 데이터 생성
skim(abalone) |>
  select(-contains("character.min"), -contains("character.max"),
         -contains("character.empty"), -contains("character.n_unique"),
         -contains("character.whitespace"), -contains("complete_rate"),
         -contains("skim_type"), -contains("numeric.hist")) |>
  flextable() |>
  highlight(i = 4, j = 5) |>
  autofit()
```

skim_variable	n_missing	numeric.mean	numeric.sd	numeric.p0	numeric.p25	numeric.p50	numeric.p75	numeric.p100
Sex	0							
Length	0	0.5239921	0.12009291	0.0750	0.4500	0.5450	0.615	0.8150
Diameter	0	0.4078813	0.09923987	0.0550	0.3500	0.4250	0.480	0.6500
Height	0	0.1395164	0.04182706	0.0000	0.1150	0.1400	0.165	1.1300
Whole_weight	0	0.8287422	0.49038902	0.0020	0.4415	0.7995	1.153	2.8255
Shucked_weight	0	0.3593675	0.22196295	0.0010	0.1860	0.3360	0.502	1.4880
Viscera_weight	0	0.1805936	0.10961425	0.0005	0.0935	0.1710	0.253	0.7600
Shell_weight	0	0.2388309	0.13920267	0.0015	0.1300	0.2340	0.329	1.0050
Rings	0	9.9336845	3.22416903	1.0000	8.0000	9.0000	11.000	29.0000

직접적인 결측치는 없지만, Height의 최소값이 0인 것과 전체적으로 데이터가 치우친 게 눈에 띈다.

결측치 및 이상치 처리

```
abalone |>
  filter(Height == 0) |>
  flextable() |>
  highlight(j = 4) |>
  autofit()
```

Sex	Length	Diameter	Height	Whole_weight	Shucked_weight	Viscera_weight	Shell_weight	Rings
I	0.430	0.34	0	0.428	0.2065	0.0860	0.1150	8
I	0.315	0.23	0	0.134	0.0575	0.0285	0.3505	6

결측치에 해당하는 observation의 데이터들이 대체적으로 Infant의 평균에 위치 하므로 평균값으로 대체하는 것이 타당하다고 여겨짐

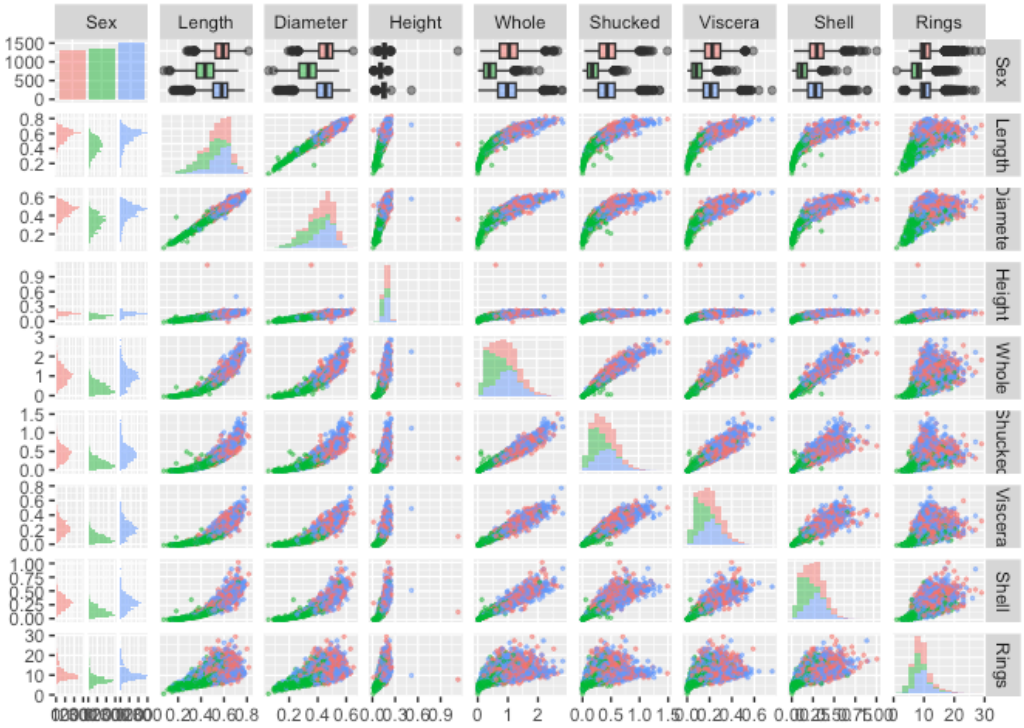
```
abalone |>
  group_by(Sex) |>
  summarise(mean_height = mean(Height)) |>
  flextable() |>
```

```
  highlight(i = 2) |>
  autofit()
```

Sex	mean_height
F	0.1580107
I	0.1079955
M	0.1513809

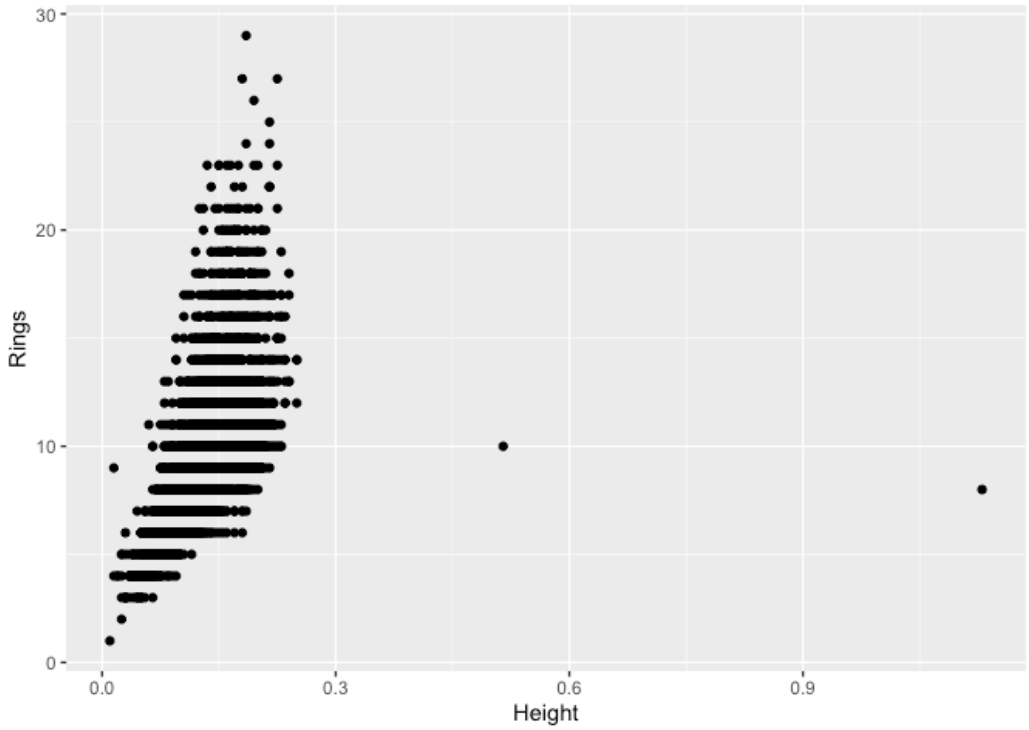
```
# 평균값 대체
ind <- which(abalone$Height == 0)
abalone$Height[ind] <- rep(.1079955, 2)
```

```
abalone |>
  ggpairs(mapping = aes(color = Sex, alpha = 0.5),
          upper = list(continuous = wrap("points", size = .5, alpha = 0.5)),
          lower = list(continuous = wrap("points", size = .5, alpha = 0.5)),
          diag = list(continuous = wrap("barDiag", bins = 15, alpha = 0.5)),
          columnLabels = c("Sex", "Length", "Diameter", "Height", "Whole", "Shucke
```



Height 열에 이상치가 존재하는 것처럼 보이고, 처리가 필요해 보인다.

```
ggplot(mapping = aes(x = Height, y = Rings), data = abalone)+
  geom_point()
```



추세에서 많이 벗어난 큰 값들은 이상치를 제거해주는게 좋아보인다.

```
abalone <- abalone |> filter(Height < .3)
```

```
abalone |>
  select_if(is.numeric) |>
  summarise_all(list(skewness = skewness)) |>  ## e1071의 skewness
  gather(Features, Skewness) |>  # tidyr의 gather
  arrange(desc(Skewness)) |>
  flextable() |>
  highlight(j = 2) |>
  autofit()
```

Features	Skewness
Rings_skewness	1.1128148
Shucked_weight_skewness	0.7144889
Shell_weight_skewness	0.6204779
Viscera_weight_skewness	0.5894320
Whole_weight_skewness	0.5283524
Height_skewness	-0.2481661
Diameter_skewness	-0.6093853
Length_skewness	-0.6399633

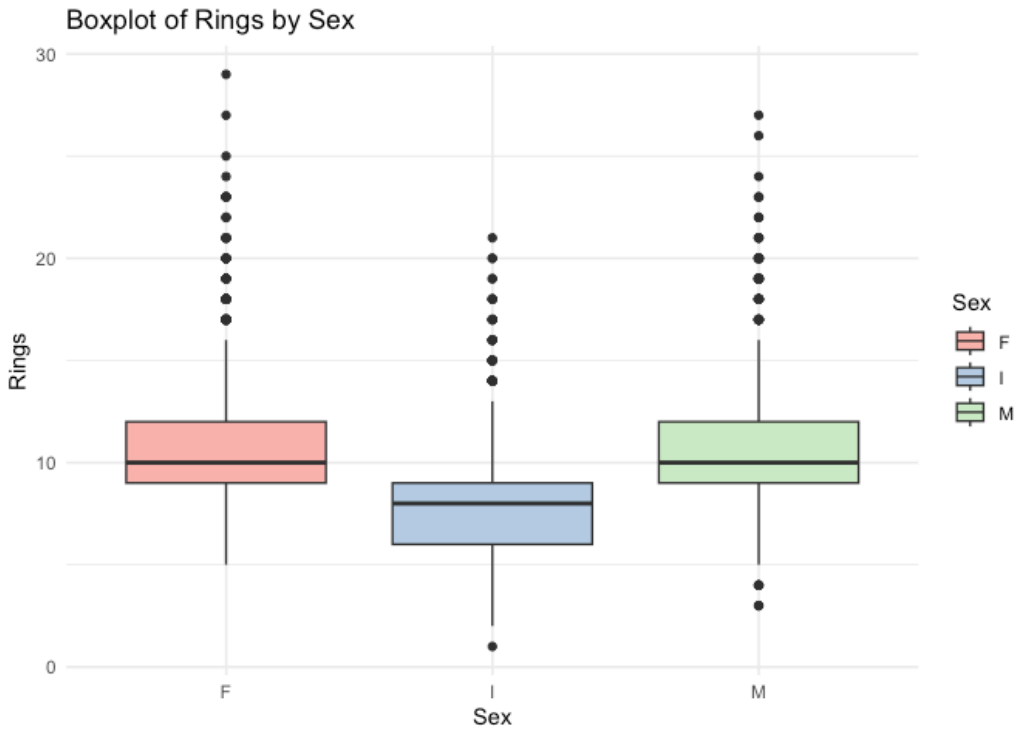
데이터 전체적으로 치우침이 존재한다.

EDA

Sex

성별 간의 나이 분포 차이가 크다면, 무게나 크기와 같은 변수에 나이 변수가 영향을 더 많이 끼칠테니, 필요에 따라서 나이 변수의 영향을 제거한 뒤 잔차를 활용해야할 수도 있다. 따라서 성별 간의 나이 분포를 우선 보는 것이 필요하다.

```
ggplot(abalone, aes(x = Sex, y = Rings, fill = Sex)) +
  geom_boxplot() +
  scale_fill_brewer(palette = "Pastel1") +
  theme_minimal() +
  labs(title = "Boxplot of Rings by Sex",
       x = "Sex",
       y = "Rings")
```




```
abalone |>
  group_by(Sex) |>
  summarise(count = n(),
    mean = mean(Rings, na.rm = TRUE),
    median = median(Rings, na.rm = TRUE),
    sd = sd(Rings, na.rm = TRUE)) |>
  flextable() |>
  highlight(i = c(1, 3), j = c(3, 5)) |>
  autofit()
```

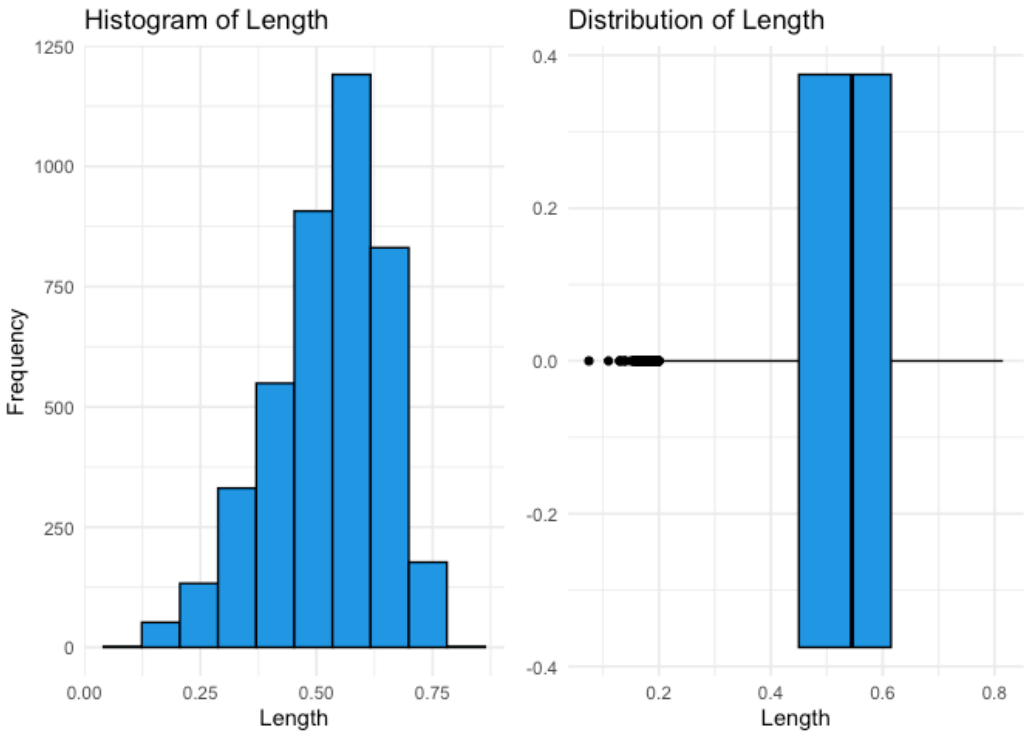
Sex	count	mean	median	sd
F	1,306	11.131700	10	3.104236
I	1,342	7.890462	8	2.511554
M	1,527	10.705959	10	3.027287

남녀 범주 별로 Rings에 대한 차이가 없어 보인다. 단, 남자 데이터의 양이 비교적 크기 때문에, 범주 간의 비교가 필요할 때는 Histogram 보다는 Density plot을 활용하는 것이 좋아 보인다.

Length

```
# 첫 번째 플롯: 히스토그램
p1 <- ggplot(abalone, aes(x = Length)) +
  geom_histogram(fill = "4", color = "black", bins = 10) +
  theme_minimal() +
  labs(title = "Histogram of Length",
```

```
       x = "Length",
       y = "Frequency")

# 두 번째 플롯: 박스 플롯
p2 <- ggplot(abalone, aes(y = Length)) +
  geom_boxplot(fill = "4", color = "black") +
  theme_minimal() +
  labs(title = "Distribution of Length",
       y = "Length") +
  coord_flip()

# 두 개의 플롯을 한 그리드에 배치
grid.arrange(p1, p2, ncol = 2)
```




```
summary(abalone$Length)
```

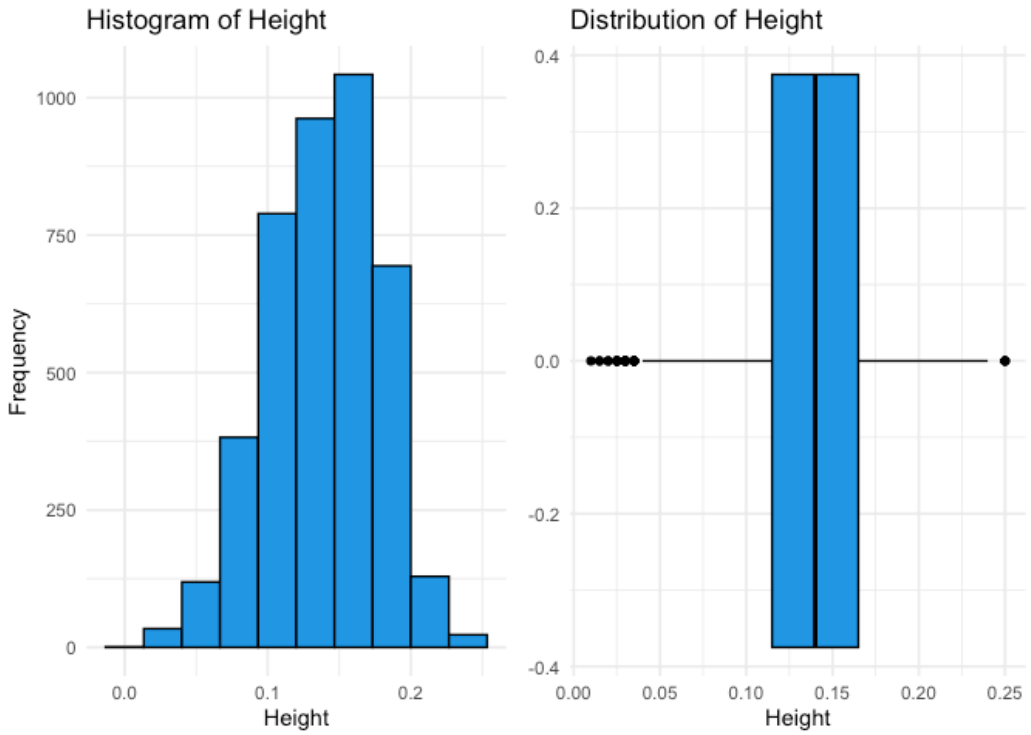
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.075   0.450   0.545   0.524   0.615   0.815
```

이상치라기엔 작은 개체의 데이터로 보인다.

Height

```
# 첫 번째 플롯: 히스토그램
p1 <- ggplot(abalone, aes(x = Height)) +
  geom_histogram(fill = "4", color = "black", bins = 10) +
  theme_minimal() +
  labs(title = "Histogram of Height",
       x = "Height",
       y = "Frequency")

# 두 번째 플롯: 박스 플롯
p2 <- ggplot(abalone, aes(y = Height)) +
  geom_boxplot(fill = "4", color = "black") +
```

```
  theme_minimal() +
  labs(title = "Distribution of Height",
       y = "Height") +
  coord_flip()

# 두 개의 플롯을 한 그리드에 배치
grid.arrange(p1, p2, ncol = 2)
```

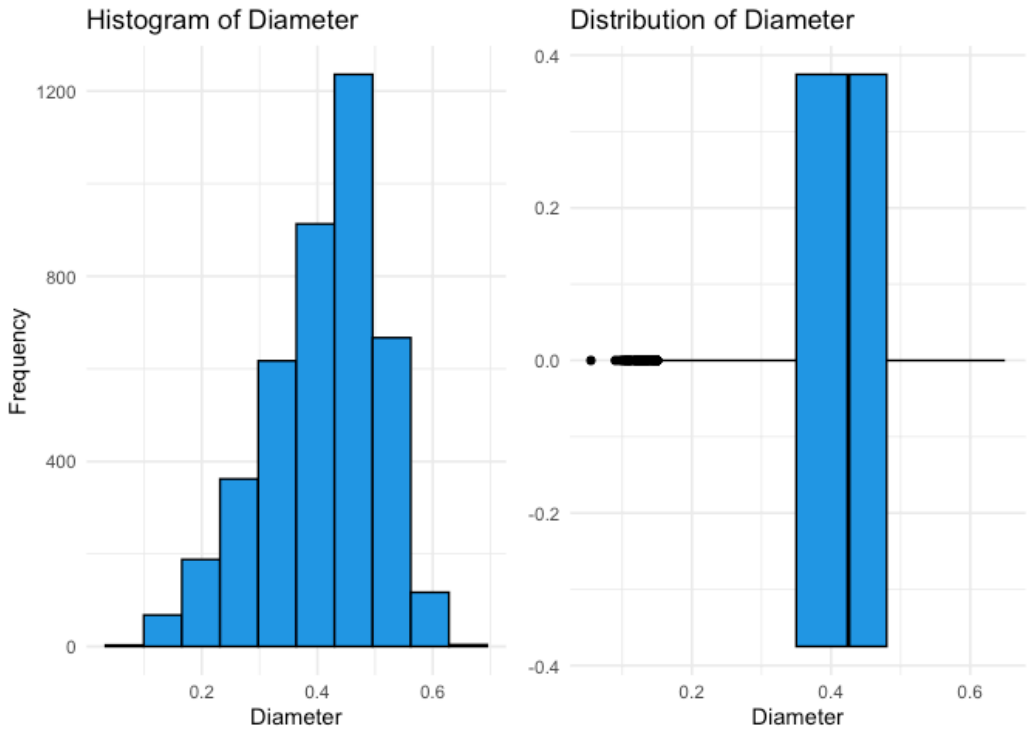



```
summary(abalone$Height)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0100  0.1150  0.1400  0.1392  0.1650  0.2500
```

Diameter

```
# 첫 번째 플롯: 히스토그램
p1 <- ggplot(abalone, aes(x = Diameter)) +
  geom_histogram(fill = "4", color = "black", bins = 10) +
  theme_minimal() +
  labs(title = "Histogram of Diameter",
       x = "Diameter",
       y = "Frequency")

# 두 번째 플롯: 박스 플롯
p2 <- ggplot(abalone, aes(y = Diameter)) +
  geom_boxplot(fill = "4", color = "black") +
  theme_minimal() +
  labs(title = "Distribution of Diameter",
       y = "Diameter") +
  coord_flip()

# 두 개의 플롯을 한 그리드에 배치
grid.arrange(p1, p2, ncol = 2)
```




```
summary(abalone$Diameter)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0550  0.3500  0.4250  0.4079  0.4800  0.6500
```

```
abalone |>
    filter(Length < 0.1 | Diameter < 0.1 | Height < 0.01) |>
  flextable() |>
  autofit()
```

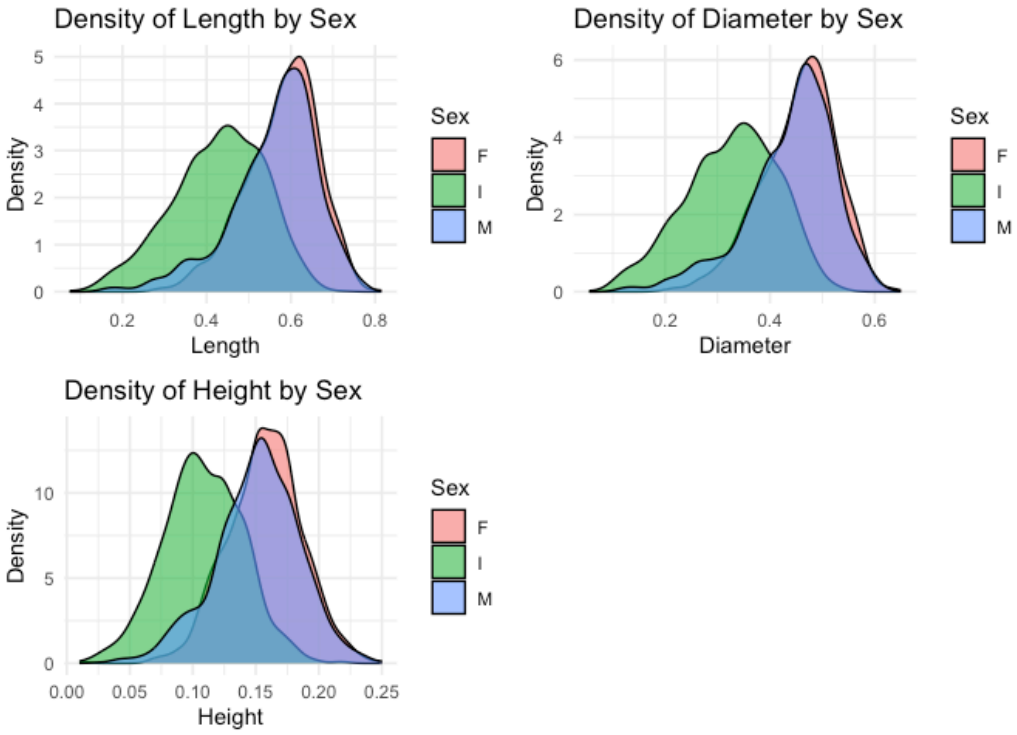
Sex	Length	Diameter	Height	Whole_weight	Shucked_weight	Viscera_weight	Shell_weight	Rings
I	0.075	0.055	0.010	0.0020	0.0010	0.0005	0.0015	1
I	0.110	0.090	0.030	0.0080	0.0025	0.0020	0.0030	3
I	0.130	0.095	0.035	0.0105	0.0050	0.0065	0.0035	4

이상치라기엔 전체적으로 작은 개체의 데이터라고 보인다.

```
# Density plot function
plot_density <- function(data, weight_var, fill_var) {
  ggplot(data, aes(x = !!sym(weight_var), fill = !!sym(fill_var))) +
    geom_density(alpha = 0.6) +
    theme_minimal() +
    labs(title = paste("Density of", weight_var, "by", fill_var),
         x = weight_var,
         y = "Density")
}
```

Density를 출력하는 함수를 선언해준다.

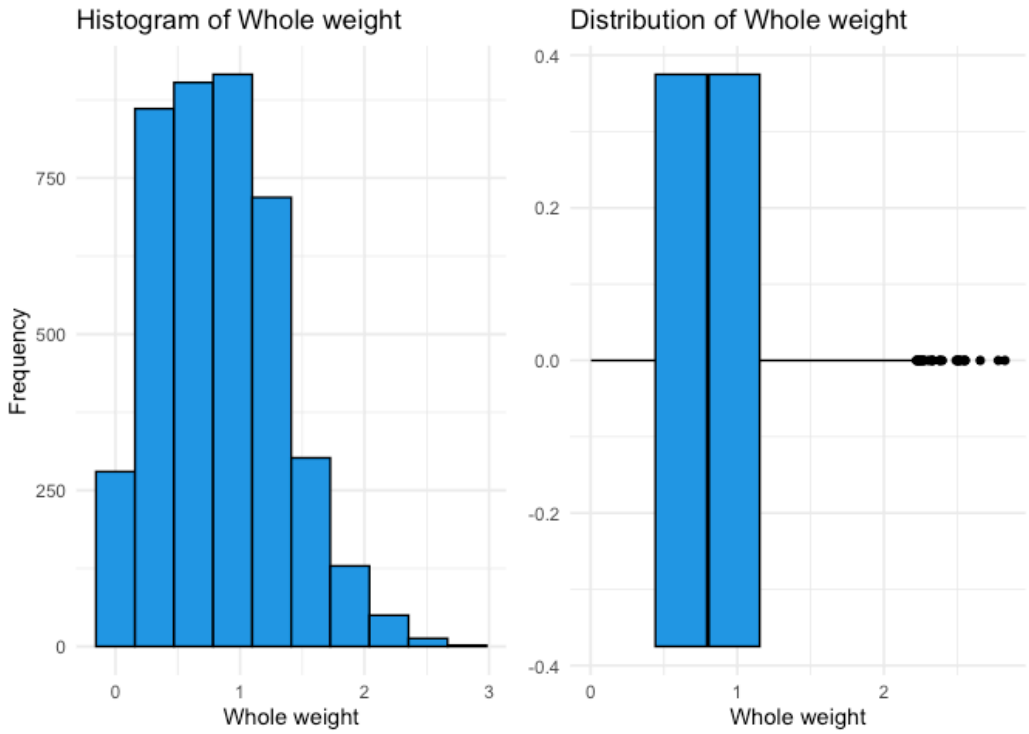
```
#  Density plot 그리기
p1 <- plot_density(abalone, "Length", "Sex")
p2 <- plot_density(abalone, "Diameter", "Sex")
p3 <- plot_density(abalone, "Height", "Sex")
grid.arrange(p1, p2, p3, ncol = 2)
```



데이터의 밀도가 전체적으로 성별에 영향을 미치지 않는 것으로 보이고, 따라서 앞선 세 개의 데이터로 성별을 예측하거나 분류하기에 어려움이 있다고 보인다.

Whole weight

```
# 첫 번째 플롯: 히스토그램
p1 <- ggplot(abalone, aes(x = Whole_weight)) +
  geom_histogram(fill = "4", color = "black", bins = 10) +
  theme_minimal() +
  labs(title = "Histogram of Whole weight",
       x = "Whole weight",
       y = "Frequency")

# 두 번째 플롯: 박스 플롯
p2 <- ggplot(abalone, aes(y = Whole_weight)) +
  geom_boxplot(fill = "4", color = "black") +
  theme_minimal() +
  labs(title = "Distribution of Whole weight",
       y = "Whole weight") +
  coord_flip()

# 두 개의 플롯을 한 그리드에 배치
grid.arrange(p1, p2, ncol = 2)
```




```
summary(abalone$Whole_weight) # min max의 차이가 크다.
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0020  0.4415  0.7995  0.8285  1.1530  2.8255
```

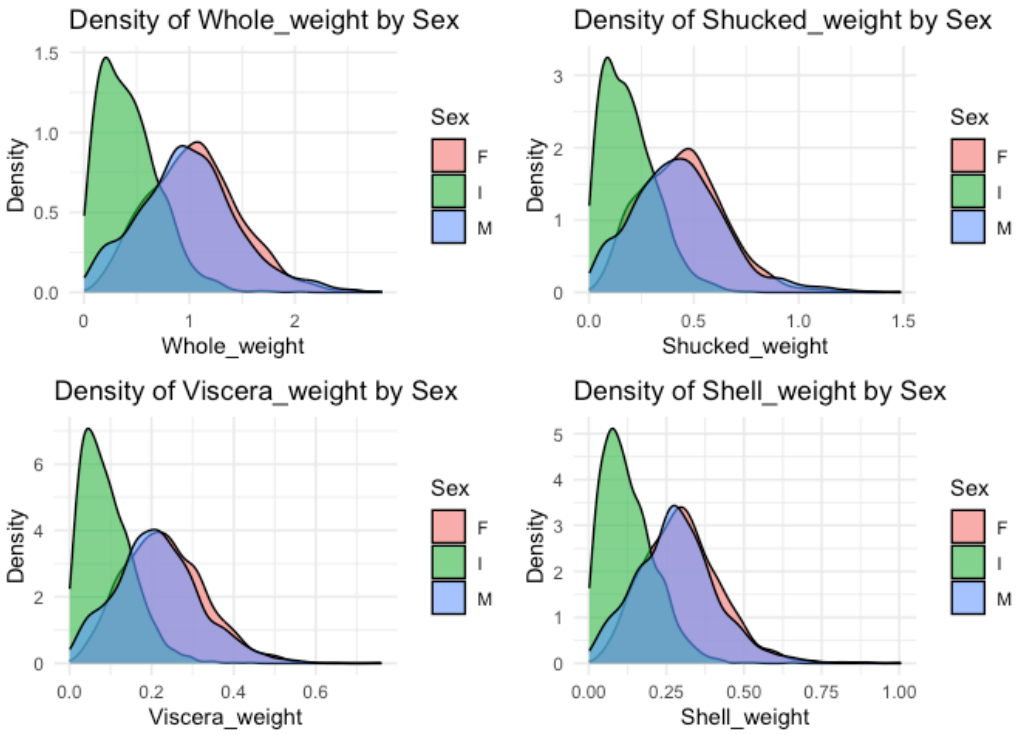
데이터의 평균에 벗어나는 큰 값이 보이고, 나이가 많은 데이터인지 성별이 서로 다른 데이터인지 확인이 필요해보인다.

```
abalone |>
    filter(Whole_weight > 2.5) |>
  flextable() |>
  highlight(j = 5) |>
  autofit()
```

Sex	Length	Diameter	Height	Whole_weight	Shucked_weight	Viscera_weight	Shell_weight	Rings
M	0.725	0.570	0.190	2.5500	1.0705	0.4830	0.7250	14
M	0.730	0.595	0.230	2.8255	1.1465	0.4190	0.8970	17
F	0.735	0.600	0.220	2.5550	1.1335	0.4400	0.6000	11
F	0.755	0.625	0.210	2.5050	1.1965	0.5130	0.6785	11
F	0.780	0.630	0.215	2.6570	1.4880	0.4985	0.5860	11
F	0.750	0.610	0.235	2.5085	1.2320	0.5190	0.6120	14
M	0.770	0.620	0.195	2.5155	1.1155	0.6415	0.6420	12
M	0.775	0.630	0.250	2.7795	1.3485	0.7600	0.5780	12
F	0.800	0.630	0.195	2.5260	0.9330	0.5900	0.6200	23
M	0.780	0.600	0.210	2.5480	1.1945	0.5745	0.6745	11

무게가 큰 데이터가 남성 데이터라는 보장이 없다. 10개의 데이터 중 5개가 남성 5개가 여성이다. 또한, Rings의 경우 11 이상의 데이터로 10 정도 이상 일때 무게가 무거워 지는 시기인 것으로 추측되는 자료를 얻을 수 있다.

```
# Apply function
p1 <- plot_density(abalone, "Whole_weight", "Sex")
p2 <- plot_density(abalone, "Shucked_weight", "Sex")
p3 <- plot_density(abalone, "Viscera_weight", "Sex")
p4 <- plot_density(abalone, "Shell_weight", "Sex")

grid.arrange(p1, p2, p3, p4, ncol = 2)
```



무게도 전체적으로 성별의 차이가 없고, 우리가 가진 변수들로 성별을 예측하기는 어려울 것 같다는 결론을 내릴 수 있다.

무게의 차이에 대한 정보

- Whole weight : 전체 무게
- Shucked weight (껍질 벗긴 무게): 전복의 고기 부분의 무게
- Viscera weight (내장 무게): 전복의 내장 무게 (출혈 후)
- Shell weight (껍질 무게): 건조 후의 껍질 무게

  따라서, Whole - (Shucked + Viscera + Shell) > 0 임을 만족해야 한다는 정보를 얻을 수 있으므로, 다음의 정보를 확인해보는 것이 필요하다.

```
diff <- abalone |>
  mutate(weight_diff = Whole_weight - (Shell_weight + Shucked_weight + Viscera_wei

summary(diff$weight_diff)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -0.44750  0.01800  0.03700  0.04995  0.06800  0.60800
```

```
diff |>
```

```
    filter(weight_diff < 0)
```

```
## # A tibble: 155 × 10
##    Sex   Length Diameter Height Whole_weight Shucked_weight Viscera_weight
##    <chr>  <dbl>    <dbl>  <dbl>        <dbl>          <dbl>          <dbl>
##  1 I      0.24     0.175  0.045        0.07           0.0315         0.0235
##  2 I      0.205    0.15   0.055        0.042          0.0255         0.015
##  3 I      0.21     0.15   0.05         0.042          0.0175         0.0125
##  4 I      0.39     0.295  0.095        0.203          0.0875         0.045
##  5 M      0.47     0.37   0.12         0.580          0.293          0.227
##  6 M      0.45     0.345  0.105        0.412          0.18           0.112
##  7 M      0.505    0.405  0.11         0.625          0.305          0.16
##  8 F      0.435    0.395  0.105        0.364          0.136          0.098
##  9 M      0.465    0.36   0.105        0.431          0.172          0.107
## 10 I      0.36     0.28   0.08         0.176          0.081          0.0505
## # ℹ 145 more rows
## # ℹ 3 more variables: Shell_weight <dbl>, Rings <dbl>, weight_diff <dbl>
```

0보다 값이 작은 데이터가 155건이나 있고, 심지어 그 크기도 큰 편이다.

```
abalone |>
  filter(Whole_weight < Shell_weight | Whole_weight < Shucked_weight | Whole_weigh
  flextable() |>
    highlight(j = 5) |>
    highlight(i = c(1,2,3,4), j = 6) |>
    highlight(i = 5, j = 8) |>
  autofit()
```

Sex	Length	Diameter	Height	Whole_weight	Shucked_weight	Viscera_weight	Shell_weight	Rings
I	0.310	0.225	0.0700000	0.1055	0.4350	0.0150	0.0400	5
I	0.275	0.205	0.0700000	0.1055	0.4950	0.0190	0.0315	5
I	0.475	0.365	0.1000000	0.1315	0.2025	0.0875	0.1230	7
I	0.355	0.270	0.0750000	0.2040	0.3045	0.0460	0.0595	7
I	0.315	0.230	0.1079955	0.1340	0.0575	0.0285	0.3505	6

해당되는 데이터 중 특정 무게가 전체 무게보다 큰 데이터들은 문제가 있어보이니, 이 데이터들의 경우 이상치로 판단하고 제거해줍니다.

```
abalone <- abalone |>
  filter(!(Whole_weight < Shell_weight | Whole_weight < Shucked_weight | Whole_wei
```

그 외의 데이터는 자세한 데이터 수집 경로 및 측정 오차에 대한 정보가 없으므로, 조작 없이 사용하도록 합니다.

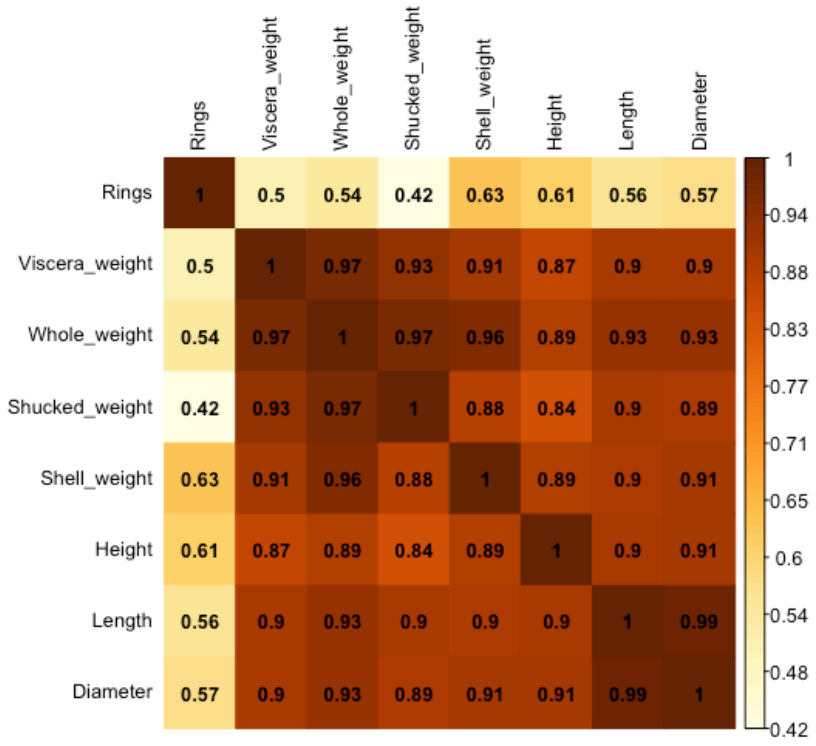
```
dim(abalone)
```

```
## [1] 4170    9
```

Height 열의 이상치 2개와 무게의 차이의 이상치 5개를 제거하여 총 7개의 데이터를 제거해주었습니다.

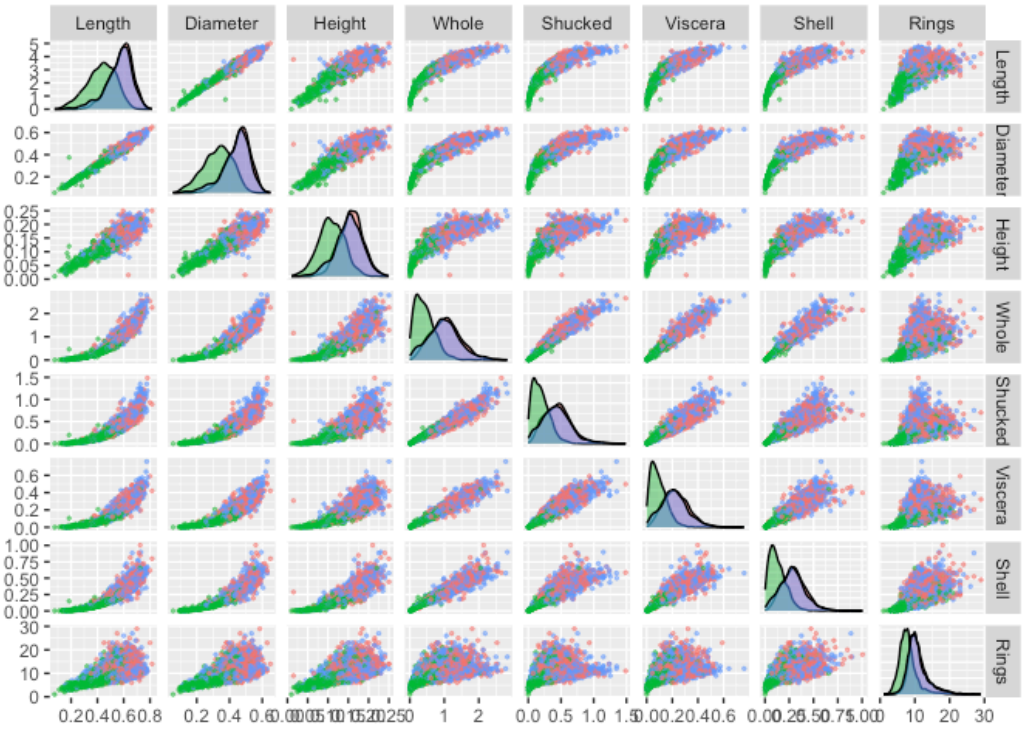
Heatmap of correlation matrix

```
# Heat map of correlation matrix
corrplot(abalone |> select_if(is.numeric) |> cor(),
  insig = "blank",
  tl.cex = .8, # 텍스트 크기
  tl.col = "black",
  method = "color",
  order = "hclust",
  addCoef.col = "black", # 계수 색상
  number.cex = .8, # 계수 텍스트 크기
  type = "full", # 모두 표시
  is.corr = F # 상관 행렬이 input이기 때문에 값의 변화는 없음.
)
```



피쳐 간의 높은 상관계수가 눈에 띄고, 특히 Length 와 Diameter 간의 상관관계와 무게들 간의 상관관계가 높은 것을 확인할 수 있다.

```
sex <- abalone$Sex

abalone |>
  select(where(is.numeric)) |>
  ggpairs(
    mapping = aes(color = sex, alpha = 0.5),
    upper = list(continuous = wrap("points", size = .5, alpha = 0.5)),
    lower = list(continuous = wrap("points", size = .5, alpha = 0.5)),
    diag = list(continuous = wrap("densityDiag", alpha = 0.5)),
    columnLabels = c("Length", "Diameter", "Height", "Whole", "Shucked", "Viscera"
  )
```



- EDA의 결과

    1. 성별에 따른 차이가 보이지 않는다.
    2. 무게는 제곱에 비례하므로 큰 개체일수록 무게가 quadratic하게 크게 측정된다.
    3. 변수 간의 다중공선성이 의심된다.

Pairwise plot

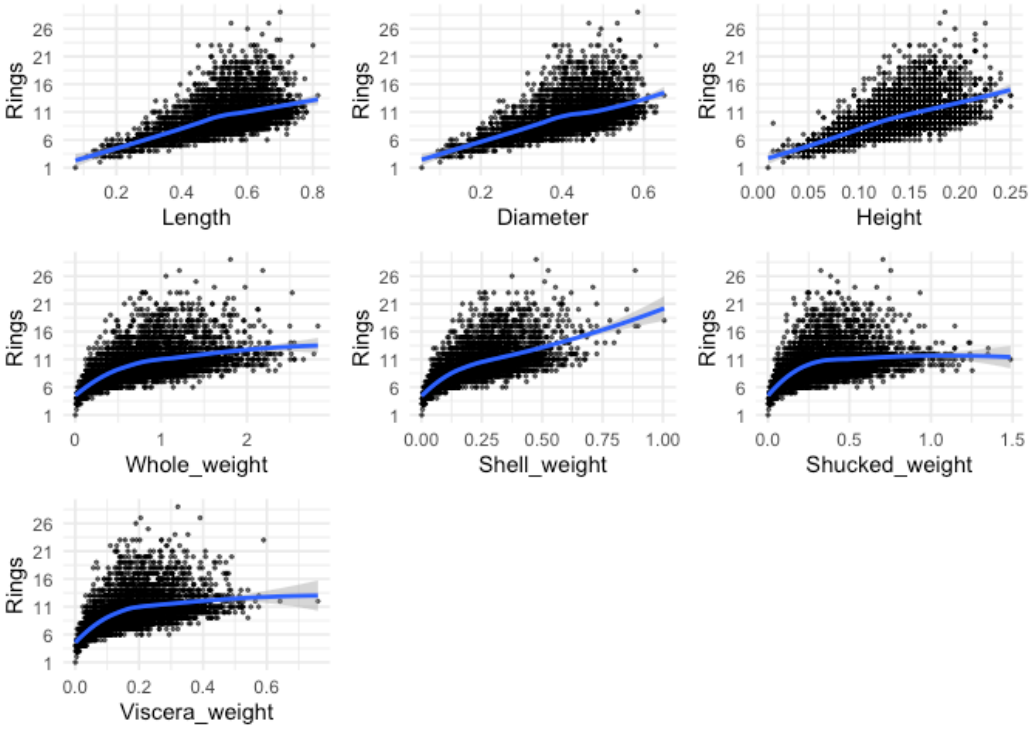
```
plot_with_loess <- function(df, x_var, y_var = "Rings") {
  ggplot(df, aes_string(x = x_var, y = y_var)) +
    geom_point(size = .5, alpha = .6) +
    geom_smooth(method = "loess") +
    scale_y_continuous(breaks = seq(min(df[[y_var]]), max(df[[y_var]]), by = 5)) +
    theme_minimal()
}
```

```
p1 <- plot_with_loess(abalone, "Length")
```

```
## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## ℹ Please use tidy evaluation idioms with `aes()`.
## ℹ See also `vignette("ggplot2-in-packages")` for more information.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
p2 <- plot_with_loess(abalone, "Diameter")
p3 <- plot_with_loess(abalone, "Height")
p4 <- plot_with_loess(abalone, "Whole_weight")
p5 <- plot_with_loess(abalone, "Shell_weight")
p6 <- plot_with_loess(abalone, "Shucked_weight")
p7 <- plot_with_loess(abalone, "Viscera_weight")

grid.arrange(p1, p2, p3, p4, p5, p6, p7, ncol = 3)
```

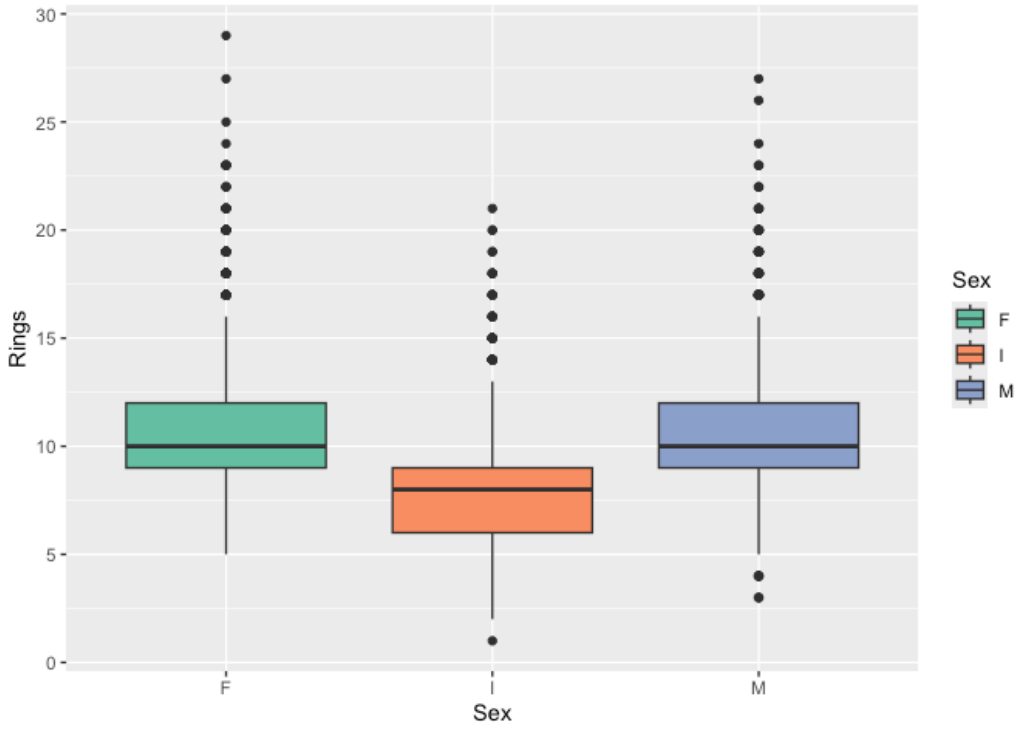
```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~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



- Length와 Diameter의 plot이 상당히 유사해 보인다.
- Weight 중에는 Shell weight이 가장 유용할 것으로 보인다.
- 전체적으로 5 ~ 12 사이에서는 Rings의 경향성 파악이 가능할 것으로 보인다.

범주 데이터 처리

```
ggplot(abalone, aes(x = Sex, y = Rings, fill = Sex)) +
  geom_boxplot() +
  scale_fill_brewer(palette = "Set2") +
  scale_y_continuous(breaks = seq(0, 30, by = 5))
```



앞서 보았듯이 Rings와 Sex 간에는 유의미한 관계가 없는 것으로 보이므로 Infant 열의 정보만을 남겨주도록 한다.

```
abalone_s <- abalone |>
  mutate(SexInfant = ifelse(Sex == "I", 1, 0)) |>
  select(-Sex)

head(abalone_s) |>
  flextable() |>
  autofit()
```

Length	Diameter	Height	Whole_weight	Shucked_weight	Viscera_weight	Shell_weight	Rings	SexInfant
0.455	0.365	0.095	0.5140	0.2245	0.1010	0.150	15	0
0.350	0.265	0.090	0.2255	0.0995	0.0485	0.070	7	0
0.530	0.420	0.135	0.6770	0.2565	0.1415	0.210	9	0
0.440	0.365	0.125	0.5160	0.2155	0.1140	0.155	10	0
0.330	0.255	0.080	0.2050	0.0895	0.0395	0.055	7	1
0.425	0.300	0.095	0.3515	0.1410	0.0775	0.120	8	1

Infant의 경우 나이에 대한 직접적인 정보를 가지고 있다는 것이 유의할 점이다. 단, 우리의 목표는 Rings에 대한 예측이므로 가지고 있는 데이터를 모두 활용하는 쪽으로 선택했다.

VIF를 통한 다중공선성 테스트

VIF 테스트는 R의 car library 함수에서 제공 중에 있어서, 회귀 모델에 적합시키면 VIF의 값이 출력된다. 본 문서에서는 자세한 결과는 기술하지 않고, 요약에 대한 부분만을 언급하도록 하겠습니다.

- Whole 에 대한 영향도 Shuked > Shell > Viscera 순으로 높다.
- Weight의 어떤 변수를 제거해도 Length와 Diameter의 VIF가 낮아지지 않는다.

결과적으로 단위가 비슷한 feature간의 VIF값이 높으므로, 해석의 용이성을 위해 단위가 서로 같은 것끼리 PCA를 진행하는 것이 좋아보입니다. 따라서 Length와 Diameter, Height에 대한 PCA와 Whole,

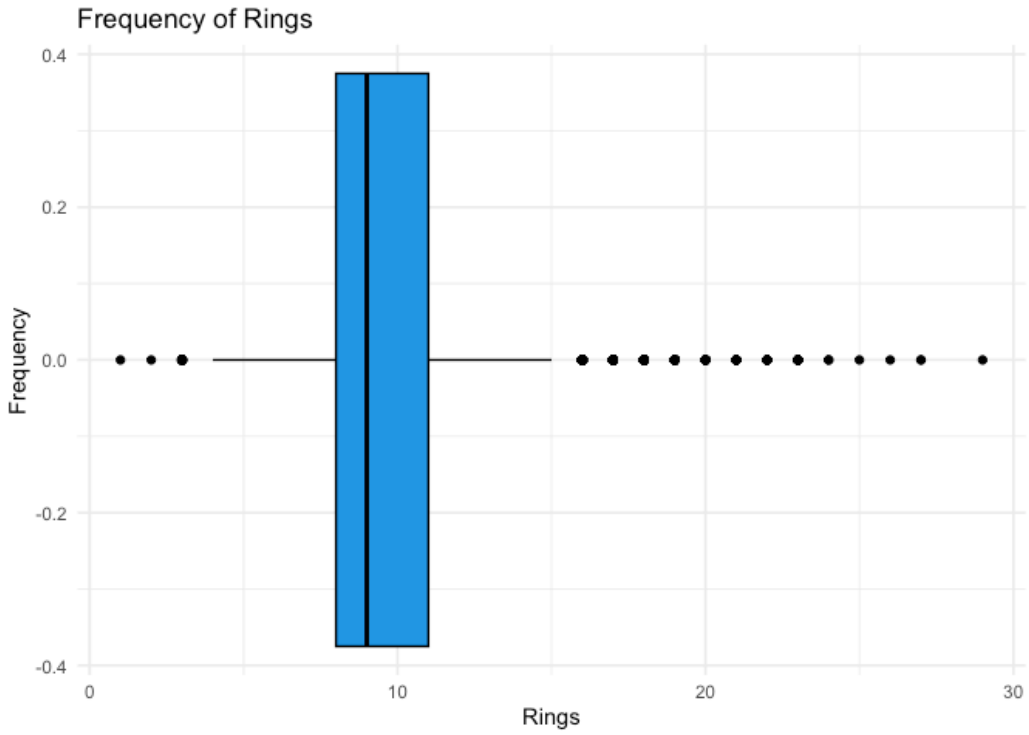
Shucked, Viscera, Shell에 대한 PCA로 다중공선성이 일으키는 문제에 대해서 해결하고자 합니다.

Rings 데이터의 분류

```
summary(abalone$Rings)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   8.000   9.000   9.939  11.000  29.000
```

```
abalone |>
  ggplot(aes(x = Rings)) +
  geom_boxplot(fill = "4", color = "black") +
  theme_minimal() +
  labs(
    title = "Frequency of Rings",
    x = "Rings",
    y = "Frequency"
  )
```



분류를 2개로 나누는 경우는 치우친 데이터의 특성상 중앙값을 기준으로 분류하는 것이 좋아보입니다. 또한 여러 개의 집단으로 분류하는 경우 1세에서 5세까지는 0, 6세에서 12세까지는 1, 13세에서 30세까지는 2로 분류합니다. (앞선 pair wise plot에서의 예측 가능성을 반영한 분류입니다.)

```
abalone |>
  select(Rings) |>
  group_by(Rings) |>
  summarise(count = n()) |>
  arrange(desc(count))
```

```
## # A tibble: 28 × 2
##    Rings count
##    <dbl> <int>
##  1     9   689
```

```
##  2    10   633
##  3     8   567
##  4    11   487
##  5     7   389
##  6    12   267
##  7     6   258
##  8    13   203
##  9    14   126
## 10     5   113
## # ℹ 18 more rows
```

모든 Rings의 값은 정수값이므로 분류는 정수값을 기준으로 진행하도록 하겠습니다.

두 집단 분류

```
class2 <- abalone_s |>
  mutate(Rings = case_when(
    Rings %in% 1:9 ~ 0,
    Rings %in% 10:30 ~ 1
  ))

write_csv(class2, "class2_basic.csv")
```

세 집단 분류

```
class3 <- abalone_s |>
  mutate(Rings = case_when(
    Rings %in% 1:5 ~ 0,
    Rings %in% 6:12 ~ 1,
    Rings %in% 13:30 ~ 2
  ))

write_csv(class3, "class3_basic.csv")
```

두 집단 분류의 경우, 이미 Infant 정보가 있기에 유의미하지 않을 수도 있습니다. 또한 5 ~ 12의 Rings가 선형관계를 잘 설명할 수 있으니, 세 집단의 분류도 함께 진행해주겠습니다.

```
class2 |>
  select(Rings) |>
  group_by(Rings) |>
  summarise(count = n()) |>
  arrange(desc(count)) |>
  flextable() |>
  autofit()
```

Rings	count
0	2,090
1	2,080

```
class3 |>
  select(Rings) |>
  group_by(Rings) |>
  summarise(count = n()) |>
```

```
  arrange(desc(count)) |>
  flextable() |>
  autofit()
```

Rings	count
1	3,290
2	693
0	187

목표했던 분류가 잘 이루어졌음을 볼 수 있습니다.

PCA

```
abalone_pca <- class2
# abalone_pca <- class3
```

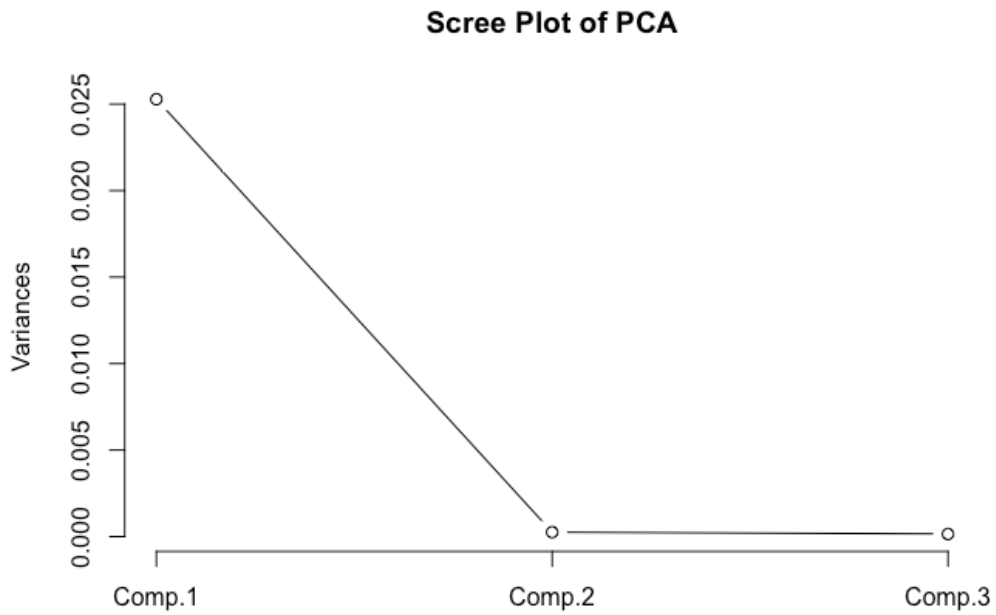
class에 따라서 주석을 해제하는 방식으로 진행하도록 한다.

LDH PCA

```
ld_pca <- princomp(abalone_pca[, c("Length", "Diameter", "Height")])
summary(ld_pca)
```

```
## Importance of components:
##                            Comp.1     Comp.2      Comp.3
## Standard deviation     0.1590007 0.01600664 0.012189144
## Proportion of Variance 0.9842409 0.00997479 0.005784284
## Cumulative Proportion  0.9842409 0.99421572 1.000000000
```

```
screeplot(ld_pca, type = "line", main = "Scree Plot of PCA")
```



높은 상관관계로 알 수 있듯이 PCA의 결과 하나의 주성분 만으로도 충분히 많은 분산을 설명할 수 있습니다.

```
abalone_pca$LD_PC1 <- ld_pca$scores[, 1]


abalone_pca <- abalone_pca |>
  select(-c("Length", "Diameter", "Height"))

head(abalone_pca)
```

```
## # A tibble: 6 × 7
##   Whole_weight Shucked_weight Viscera_weight Shell_weight Rings SexInfant
##          <dbl>          <dbl>          <dbl>        <dbl> <dbl>     <dbl>
## 1        0.514         0.224          0.101         0.15      1         0
## 2        0.226         0.0995         0.0485        0.07      0         0
## 3        0.677         0.256          0.142         0.21      0         0
## 4        0.516         0.216          0.114         0.155     1         0
## 5        0.205         0.0895         0.0395        0.055     0         1
## 6        0.352         0.141          0.0775        0.12      0         1
## # ℹ 1 more variable: LD_PC1 <dbl>
```

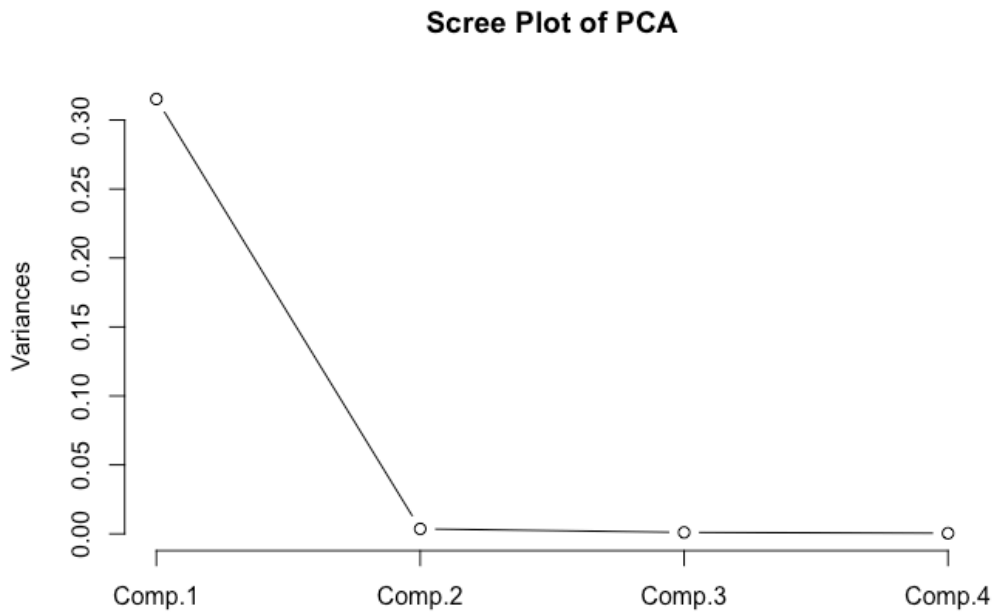
PCA의 결과를 데이터에 저장해주고, 기존의 Length, Diameter, Height의 열은 제거해줍니다.

Weight PCA

```
weight_pca <- princomp(abalone_pca[, c("Whole_weight", "Shucked_weight", "Viscera_
summary(weight_pca)
```

```
## Importance of components:
##                            Comp.1     Comp.2      Comp.3      Comp.4
## Standard deviation     0.5614451 0.05999399 0.032200713 0.020685678
## Proportion of Variance 0.9841889 0.01123775 0.003237389 0.001335991
## Cumulative Proportion  0.9841889 0.99542662 0.998664009 1.000000000
```

```
# Scree plot을 그립니다.
screeplot(weight_pca, type = "line", main = "Scree Plot of PCA")
```



마찬가지로, 높은 상관성 답게 적은 주성분으로 거의 모든 분산이 설명이 가능합니다. 최대한 적은 데이터의 손실을 위해 두 개의 주성분을 이용해주도록 하겠습니다.

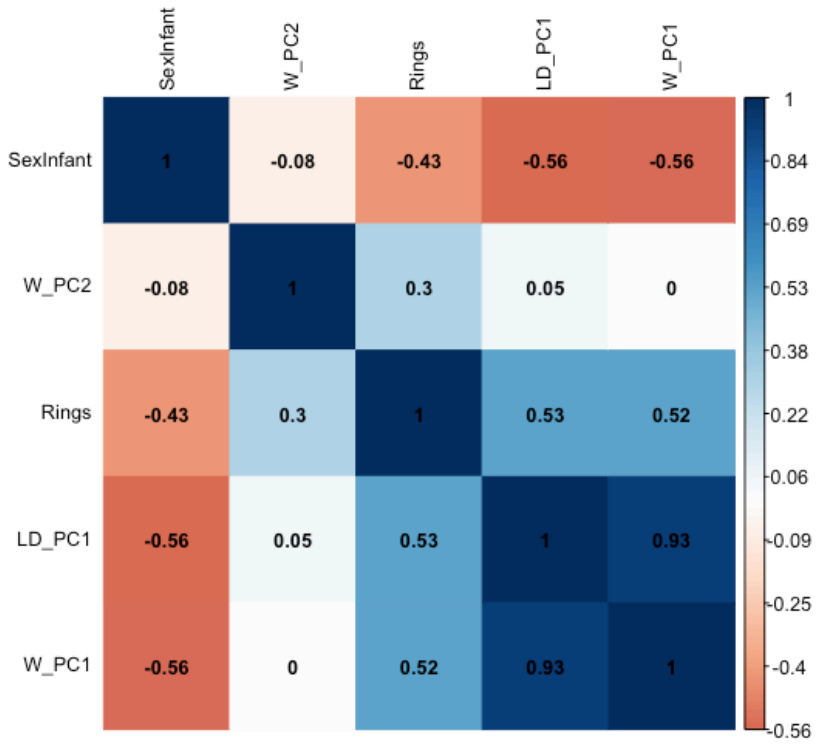
```
abalone_pca$W_PC1 <- weight_pca$scores[, 1]
abalone_pca$W_PC2 <- weight_pca$scores[, 2]

abalone_pca <- abalone_pca |>
  select(-c("Whole_weight", "Viscera_weight", "Shucked_weight", "Shell_weight"))

head(abalone_pca)
```

```
## # A tibble: 6 × 5
##   Rings SexInfant  LD_PC1  W_PC1    W_PC2
##   <dbl>     <dbl>   <dbl>  <dbl>    <dbl>
## 1     1         0 -0.0885 -0.363 -0.00606
## 2     0         0 -0.231  -0.691 -0.00891
## 3     0         0  0.0109 -0.187  0.0347
## 4     1         0 -0.0932 -0.361  0.00463
## 5     0         1 -0.254  -0.718 -0.0141
## 6     0         1 -0.151  -0.548  0.0118
```

마찬가지로 PCA의 두 주성분을 데이터 프레임에 추가해주고, 기존의 열을 제거해줍니다.

```
# Heat map of correlation matrix
corrplot(abalone_pca |> select_if(is.numeric) |> cor(),
  insig = "blank",
  tl.cex = .8, # 텍스트 크기
  tl.col = "black",
  method = "color",
  order = "hclust",
  addCoef.col = "black", # 계수 색상
  number.cex = .8, # 계수 텍스트 크기
  type = "full", # 모두 표시
```

```
  is.corr = F # 상관 행렬이 input이기 때문에 값의 변화는 없음.
)
```



결과적으로 높은 상관성의 대부분을 해결할 수 있게 되었습니다.

```
model <- lm(Rings ~ ., data = abalone_pca)
vif(model)
```

```
## SexInfant    LD_PC1     W_PC1     W_PC2
##  1.485610  7.823184  7.786917  1.026442
```

VIF의 값도 10이 넘어가던 값에서 10 미만으로 해결되었음을 볼 수 있습니다.

```
write.csv(abalone_pca, "class2_pca.csv")
```

결론

- EDA를 통해 데이터를 파악하고, 이상치 및 결측치를 제거해주었습니다.
- 다중공선성이 문제될 수 있기 때문에, PCA를 통해서 해결하고자 했습니다.

1. 2개의 범주로 나눈 “class2_basic.csv”
2. 3개의 범주로 나눈 “class3_basic.csv”
3. 2개의 범주로 나누고 PCA를 진행한 “class2_pca.csv”
4. 3개의 범주로 나누고 PCA를 진행한 “class3_pca.csv”