항공사 승객 만족도 조사 결과 분석

다변량 분석 및 실습

2조 신효진 이민채 박경숙

0. 서론

데이터 출처 - Airline Passenger Satisfaction (kaggle.com)

데이터 변수 설명 –

승객 정보

Gender: 승객의 성별 (여성, 남성)

Customer Type: 고객 유형 (단골 고객, 비단골 고객)

Age: 승객의 나이

Type of Travel: 승객의 비행 목적 (개인 여행, 비즈니스 여행)

Class: 승객이 탑승한 비행기의 클래스 (비즈니스, 에코, 에코 플러스)

Flight distance: 비행 거리

기내 만족도

Inflight wifi service: 기내 Wi-Fi 서비스 만족도 (0:해당 없음, 1~5)

Food and drink: 음식 및 음료의 만족도

Seat comfort: 좌석 편안함에 대한 만족도

Inflight entertainment: 기내 여흥 만족도

Leg room service : 레그룸 서비스(다리를 뻗을 수 있는 공간) 만족도

Inflight service: 기내 서비스 만족도

Cleanliness : 청결 만족도

기타 편의 및 항공사 만족도

Departure/Arrival time convenient : 출·도착 시간 편리성에 대한 만족도

Ease of Online booking: 온라인 예약 만족도

Gate location : Gate 위치에 대한 만족도

Online boarding : 온라인 탑승 수속 만족도

On-board service: 탑승 서비스 만족도

Baggage handling : 수하물 처리 만족도

Check-in service : 체크인 서비스 만족도

Departure Delay in Minutes: 출발 시 지연된 시간(분)

Arrival Delay in Minutes: 도착 시 지연된 시간(분)

Satisfaction : 항공사 만족도(만족, 보통, 불만족)

```
summary(df)
```

```
> summary(df)
      ...1             id            Gender          Customer Type          Age        Type of Travel       Class
 Min.   :    0   Min.   :    17   Length:25976       Length:25976       Min.   : 7.00   Length:25976       Length:25976
 1st Qu.: 6494   1st Qu.: 32171   Class :character   Class :character   1st Qu.:27.00   Class :character   Class :character
 Median :12988   Median : 65320   Mode  :character   Mode  :character   Median :40.00   Mode  :character   Mode  :character
 Mean   :12988   Mean   : 65006                                        Mean   :39.62
 3rd Qu.:19481   3rd Qu.: 97584                                        3rd Qu.:51.00
 Max.   :25975   Max.   :129877                                        Max.   :85.00

 Flight Distance Inflight wifi service Departure/Arrival time convenient Ease of Online booking Gate location   Food and drink
 Min.   :  31    Min.   :0.000         Min.   :0.000                     Min.   :0.000          Min.   :1.000   Min.   :0.000
 1st Qu.: 414    1st Qu.:2.000         1st Qu.:2.000                     1st Qu.:2.000          1st Qu.:2.000   1st Qu.:2.000
 Median : 849    Median :3.000         Median :3.000                     Median :3.000          Median :3.000   Median :3.000
 Mean   :1194    Mean   :2.725         Mean   :3.047                     Mean   :2.757          Mean   :2.977   Mean   :3.215
 3rd Qu.:1744    3rd Qu.:4.000         3rd Qu.:4.000                     3rd Qu.:4.000          3rd Qu.:4.000   3rd Qu.:4.000
 Max.   :4983    Max.   :5.000         Max.   :5.000                     Max.   :5.000          Max.   :5.000   Max.   :5.000

 Online boarding  Seat comfort   Inflight entertainment On-board service Leg room service Baggage handling Checkin service
 Min.   :0.000   Min.   :1.000   Min.   :0.000          Min.   :0.000    Min.   :0.00     Min.   :1.000    Min.   :1.000
 1st Qu.:2.000   1st Qu.:2.000   1st Qu.:2.000          1st Qu.:2.000    1st Qu.:2.00     1st Qu.:3.000    1st Qu.:3.000
 Median :4.000   Median :4.000   Median :4.000          Median :4.000    Median :4.00     Median :4.000    Median :3.000
 Mean   :3.262   Mean   :3.449   Mean   :3.358          Mean   :3.386    Mean   :3.35     Mean   :3.633    Mean   :3.314
 3rd Qu.:4.000   3rd Qu.:5.000   3rd Qu.:4.000          3rd Qu.:4.000    3rd Qu.:4.00     3rd Qu.:5.000    3rd Qu.:4.000
 Max.   :5.000   Max.   :5.000   Max.   :5.000          Max.   :5.000    Max.   :5.00     Max.   :5.000    Max.   :5.000

 Inflight service  Cleanliness    Departure Delay in Minutes Arrival Delay in Minutes satisfaction
 Min.   :0.000     Min.   :0.000   Min.   :   0.00          Min.   :   0.00          Length:25976
 1st Qu.:3.000     1st Qu.:2.000   1st Qu.:   0.00          1st Qu.:   0.00          Class :character
 Median :4.000     Median :3.000   Median :   0.00          Median :   0.00          Mode  :character
 Mean   :3.649     Mean   :3.286   Mean   :  14.31          Mean   :  14.74
 3rd Qu.:5.000     3rd Qu.:4.000   3rd Qu.:  12.00          3rd Qu.:  13.00
 Max.   :5.000     Max.   :5.000   Max.   :1128.00          Max.   :1115.00
                                                            NA's   :83
```

프로젝트 목표 –

수업시간에 배운 다양한 다변량 분석 방법을 통해 세부적인 만족도 지표들 중 승객의 항공사 만족도에 가장 큰 영향을 미치는 요소를 확인함으로써 항공사가 우선적으로 집중해야 할 부분이 무엇인지 제시하고자 한다.

가설 –

출발, 도착 시간이 지연되지 않는 것이 승객의 만족도를 높일 것이다.

➔ PCA를 통해 선택된 주성분에서 출발, 도착 지연 시간이 비교적 큰 계수를 갖는지 확인하고자 한다.

승객 만족도에 주요하게 영향을 미치는 변수 그룹이 존재할 것이다.

➔ Clustering을 통해 만족과 불만족 그룹의 차이가 명확하게 구분되는 군집을 찾고, 해당 군집의 특성을 파악하여 승객 만족도에 영향을 미치는 변수를 찾고자 한다.

1. 데이터 전처리

Departure/arrival time convenient

: 결측치(만족도가 0인 경우)가 가장 많이 존재하고, 대체 가능한 변수(출발, 도착 지연 시간)이 존재하므로 해당 변수를 자료에서 삭제하였다.

```
# 빈 벡터 생성하여 결과 저장 데이터 프레임의 각 열에 대해 반복 결과를 벡터에 저장 결과를 데이터 프레임으로 변환 함수 실행
zero_counts_df <- count_zeros(df)
print(zero_counts_df)
```

```
> zero_counts_df <- count_zeros(df)
> print(zero_counts_df)
                              Column ZeroCount
1                               ...1         1
2                                 id         0
3                             Gender         0
4                      Customer Type         0
5                                Age         0
6                      Type of Travel         0
7                              Class         0
8                    Flight Distance         0
9               Inflight wifi service       813
10 Departure/Arrival time convenient      1381
11            Ease of Online booking      1195
12                     Gate location         0
13                     Food and drink        25
14                    Online boarding       652
15                       Seat comfort         0
16            Inflight entertainment         4
17                  On-board service         2
18                  Leg room service       126
19                  Baggage handling         0
20                   Checkin service         0
21                  Inflight service         2
22                       Cleanliness         2
23        Departure Delay in Minutes     14688
24          Arrival Delay in Minutes     14594
25                      satisfaction         0
```

Arrival Delay in Minutes

: 도착 지연 시간에 대한 결측치가 83개 존재하여 지연 시간이 없는 것으로 판단하고 결측치를 0으로 대체하였다.

```
# 결측치 확인
 colSums(is.na(df))
```

```
> colSums(is.na(df))
                              ...1                                id
                                 0                                 0
                               Age                     Type of Travel
                                 0                                 0
          Inflight wifi service Departure/Arrival time convenient
                                 0                                 0
                Food and drink                    Online boarding
                                 0                                 0
              On-board service                    Leg room service
                                 0                                 0
              Inflight service                         Cleanliness
                                 0                                 0
                satisfaction
                                 0
                             Gender                     Customer Type
                                 0                                 0
                             Class                     Flight Distance
                                 0                                 0
          Ease of Online booking                     Gate location
                                 0                                 0
                     Seat comfort             Inflight entertainment
                                 0                                 0
                Baggage handling                    Checkin service
                                 0                                 0
      Departure Delay in Minutes        Arrival Delay in Minutes
                                 0                                83
```

Ease of Online booking / Online boarding / Inflight wifi service

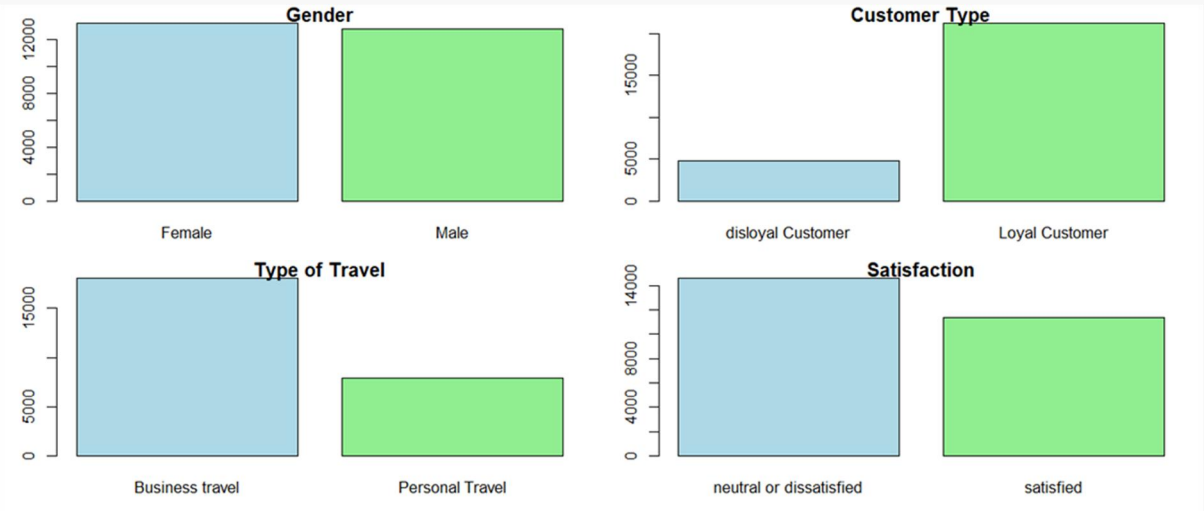
: 온라인 서비스, 기내 wifi 서비스에 대한 결측치는 해당 서비스를 이용하지 않은 승객으로 판단하고 결측치를 평균값을 대체하였다.

Gender / Customer Type / Type of Travel / Class / Satisfaction

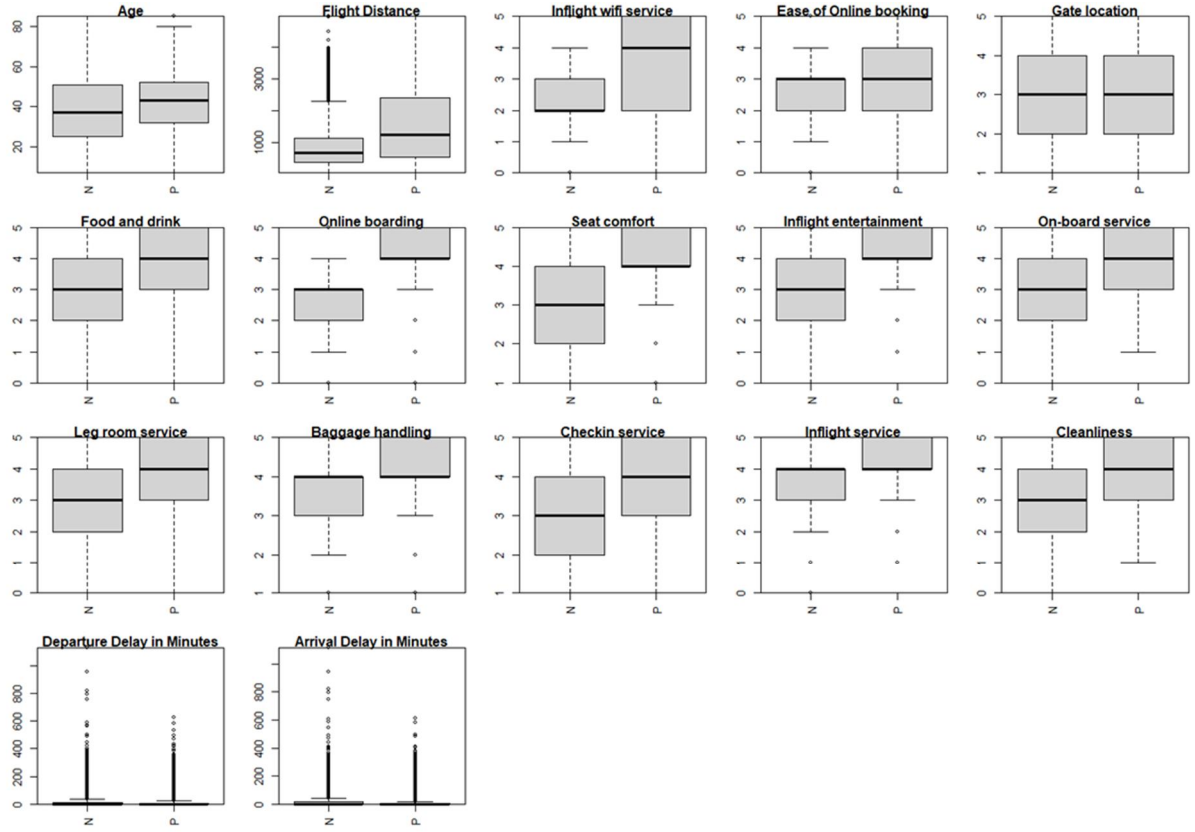
: Class가 character인 변수들의 값을 분석에 이용하기 위해 숫자로 변환하였다.

2. EDA

```r
# 각 범주형 변수의 고유한 값과 빈도 계산
 unique_gender <- table(df$Gender)
 unique_customer_type <- table(df$`Customer Type`)
 unique_travel_type <- table(df$`Type of Travel`)
 unique_satisfaction <- table(df$satisfaction)

par(mfrow = c(2, 2))

# Bar plot for Gender
 barplot(unique_gender, col = c("lightblue", "lightgreen"),
         main = "Gender", xlab = "Gender", ylab = "Count")

# Bar plot for Customer Type
 barplot(unique_customer_type, col = c("lightblue", "lightgreen"),
         main = "Customer Type", xlab = "Customer Type", ylab = "Count")

# Bar plot for Type of Travel
 barplot(unique_travel_type, col = c("lightblue", "lightgreen"),
         main = "Type of Travel", xlab = "Type of Travel", ylab = "Count")

# Bar plot for Satisfaction
 barplot(unique_satisfaction, col = c("lightblue", "lightgreen"),
         main = "Satisfaction", xlab = "Satisfaction", ylab = "Count")
```



```r
# 인덱스와 사전에 삭제하려는 컬럼을 제외한 데이터프레임 생성

df_filtered <- df[, -c(1, 2, 10)]

# 숫자형 열 이름 추출
 numerical_columns <- names(df_filtered)[sapply(df_filtered, is.numeric)]

par(mfrow = c(4, 5), mar = c(3, 3, 1, 1))

# 박스플롯을 그리는 함수 정의
 num_plotter <- function(data, target) {
   numerical_columns <- names(data)[sapply(data, is.numeric)]
   num_plots <- length(numerical_columns)
   n_cols <- 5
   n_rows <- ceiling(num_plots / n_cols)



  for (col in numerical_columns) {
     boxplot(data[[col]] ~ data[[target]], xlab = "", ylab = '', main =
col, axes = FALSE)
     box()



    # 사용자 정의 축 레이블 추가
     custom_labels <- ifelse(levels(factor(data[[target]])) %in%
c("Neutral or dissatisfied", "satisfied"), "P", "N")
     axis(side = 1, at = 1:length(custom_labels), labels = custom_labels,
las = 2)
     axis(side = 2)

  }
 }

# 박스플롯 그리기
 num_plotter(df_filtered, "satisfaction")
```



3. (1) PCA

```
# 주성분분석을 위해 숫자형 변수들만 선택 (범주형 변수 제외)
df_numeric <- df_filled[, sapply(df_filled, is.numeric)]

# 제외할 변수명 리스트
exclude_vars <- c("id", "...1", "Departure/Arrival time convenient")

# 변수명이 제외할 변수들을 제외한 복사본 생성
df_numeric <- df_numeric[, !names(df_numeric) %in% exclude_vars]


# 주성분 분석 수행
pca_result <- prcomp(df_numeric, cor = TRUE)
## Warning: In prcomp.default(df_numeric, cor = TRUE) : ##  extra argument
'cor' will be disregarded
# 주성분(PC)의 설명력 확인
summary(pca_result)
## Importance of components:
##                             PC1      PC2      PC3     PC4     PC5   PC6
PC7
## Standard deviation     998.685 52.36653 15.06439 7.87448 2.48040 1.856
1.829
```

```
## Proportion of Variance   0.997  0.00274  0.00023 0.00006 0.00001 0.000
0.000
## Cumulative Proportion    0.997  0.99969  0.99992 0.99998 0.99998 1.000
1.000
##                           PC8   PC9  PC10   PC11   PC12   PC13   PC14   P
C15
## Standard deviation     1.295 1.143 1.063 0.8896 0.8568 0.7456 0.7126
0.7015
## Proportion of Variance 0.000 0.000 0.000 0.0000 0.0000 0.0000 0.0000
0.0000
## Cumulative Proportion  1.000 1.000 1.000 1.0000 1.0000 1.0000 1.0000
1.0000
##                           PC16   PC17   PC18   PC19   PC20  PC21   PC22
## Standard deviation     0.6872 0.6076 0.5658 0.4984 0.4046 0.354 0.2314
## Proportion of Variance 0.0000 0.0000 0.0000 0.0000 0.0000 0.000 0.0000
## Cumulative Proportion  1.0000 1.0000 1.0000 1.0000 1.0000 1.000 1.0000
```

```r
# 누적 설명력 계산 및 출력
cumulative_variance <- cumsum(pca_result$sdev^2) / sum(pca_result$sdev^2)
* 100
cat("\nCumulative variance explained:\n")
```

```
##
## Cumulative variance explained:
```

```r
print(cumulative_variance)
```

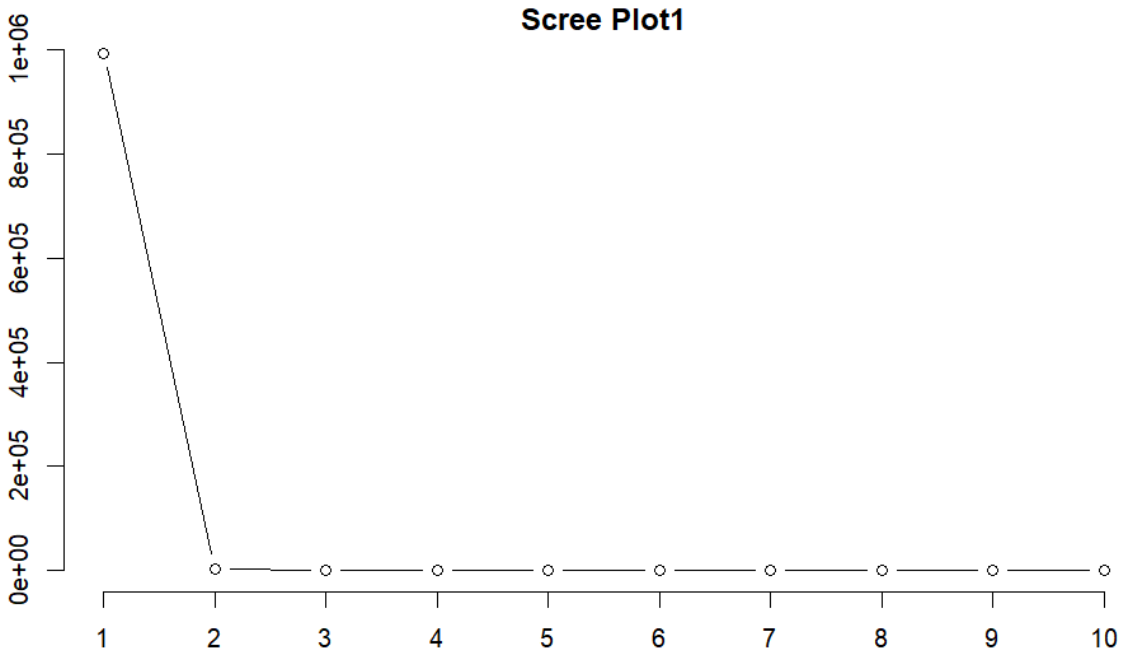
```
##  [1]  99.69482  99.96893  99.99161  99.99781  99.99843  99.99877  99.999
11
##  [8]  99.99927  99.99940  99.99952  99.99960  99.99967  99.99972  99.999
78
##
[15]  99.99982  99.99987  99.99991  99.99994  99.99997  99.99998  99.99999
## [22] 100.00000
```

```r
# PC1 의 정보 불러오기
pc1_info <- pca_result$rotation[, 1]

# 값이 높은 순서대로 정렬
pc1_info_sorted <- sort(pc1_info, decreasing = TRUE)

# 정렬된 PC1 의 정보 출력
print(pc1_info_sorted)
```

```
##            Flight Distance                        Age
##               9.999987e-01               1.506944e-03
##            Online boarding               Seat comfort
##               2.401775e-04               2.101190e-04
##     Inflight entertainment           Leg room service
##               1.850176e-04               1.717921e-04
##          On-board service               satisfaction
##               1.519444e-04               1.467394e-04
##                Cleanliness Departure Delay in Minutes
##               1.395748e-04               1.292962e-04
##            Checkin service              Customer Type
##               9.624156e-05               8.911308e-05
##          Baggage handling           Inflight service
```

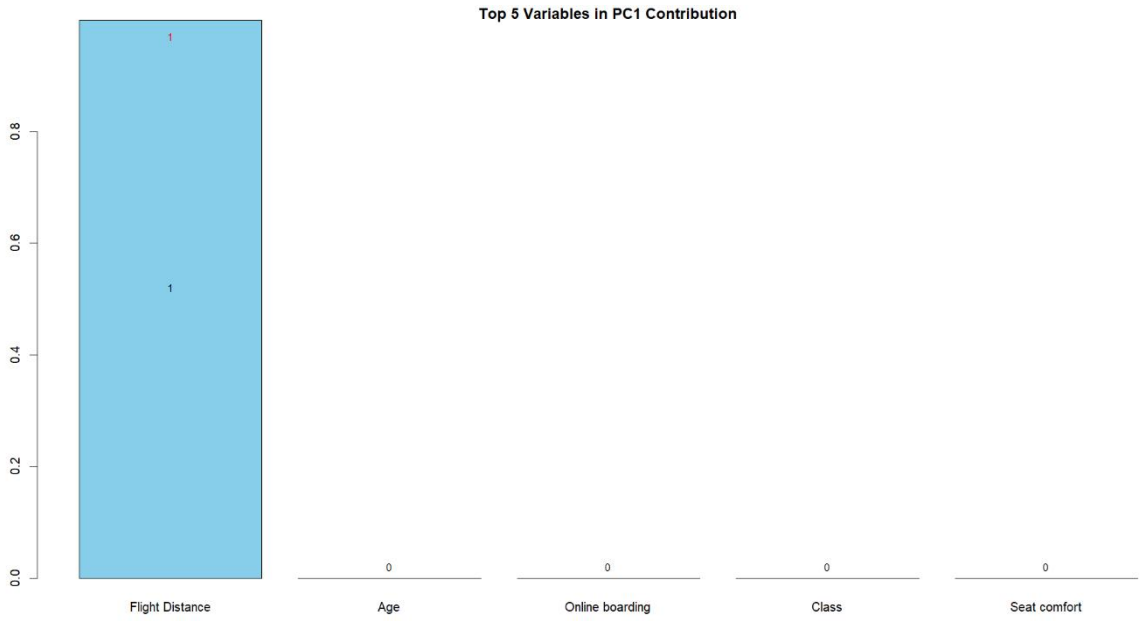
```
##                8.428960e-05                7.910478e-05
##              Food and drink     Ease of Online booking
##                7.558458e-05                5.809929e-05
##               Gate location      Inflight wifi service
##                1.079742e-05                8.982933e-06
##    Arrival Delay in Minutes                      Gender
##                4.544507e-06               -2.613894e-06
##              Type of Travel                       Class
##               -1.220339e-04               -2.658961e-04
```

```r
# ---------------------- Scree plot 그리기 ----------------------

par(mfrow = c(1, 1))
plot(pca_result, type = "l", main = "Scree Plot1")
```




```r
# ---------------------- 주성분의 변수 기여도 ----------------------

# PC1 내의 변수 기여도(회전된 값의 제곱) 계산
pc1_contributions <- (pca_result$rotation[, 1])^2

# 기여도가 높은 순서대로 상위 5 개 변수 선택
top_5_vars <- names(sort(pc1_contributions, decreasing = TRUE))[1:5]
pc1_contributions_top5 <- pc1_contributions[top_5_vars]

# 회전된 값 구하기
rotation_values <- pca_result$rotation[, 1]

# 그래프 그리기
barplot_heights <- barplot(pc1_contributions_top5, col = "skyblue",
        main = "Top 5 Variables in PC1 Contribution", xlab = "Variables",
```

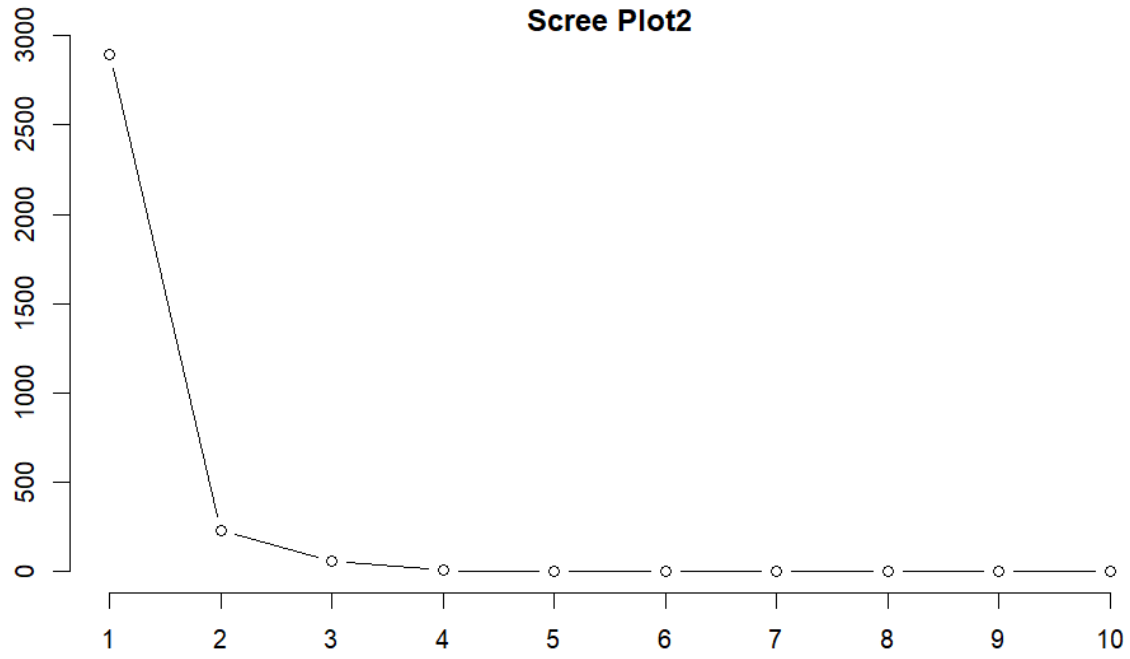
```
ylab = "Contribution")

# 각 막대 내부에 기여도 값 추가
for (i in 1:length(barplot_heights)) {
  text(barplot_heights[i], pc1_contributions_top5[i] / 2, labels =
round(pc1_contributions_top5[i], 4), pos = 3, col = "black", cex = 0.8)
}

# 각 막대 위에 회전된 값 추가 (막대의 위로 올리기 위해 적절한 y 값 설정)
text(barplot_heights, pc1_contributions_top5 - 0.05 *
diff(range(pc1_contributions_top5)),
     labels = round(rotation_values[top_5_vars], 4), pos = 3, col =
"red", cex = 0.8)
```



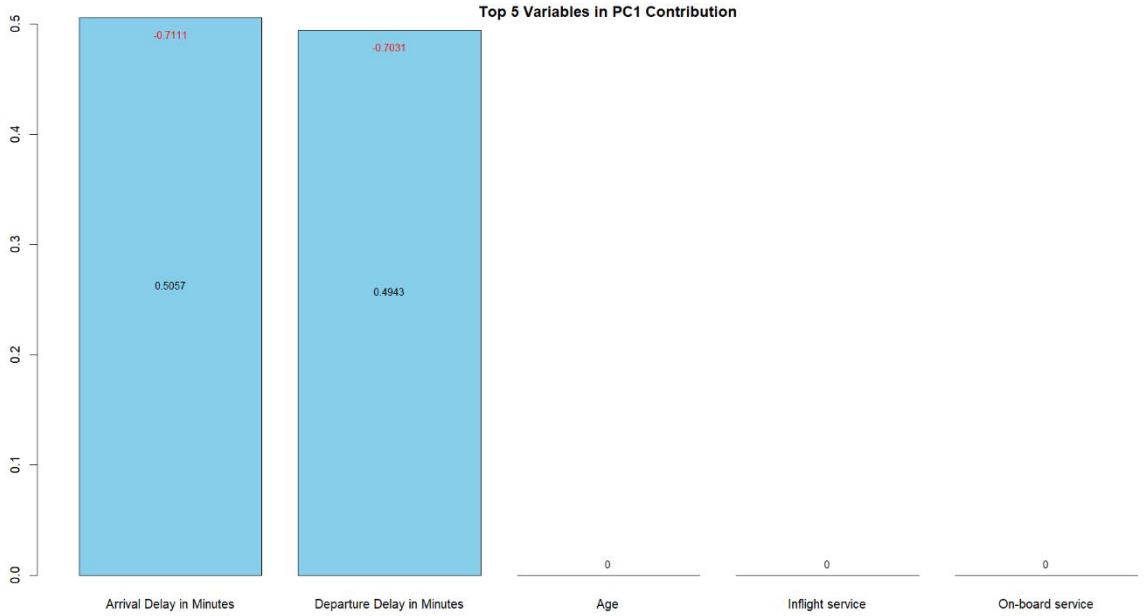
2-2. PCA(2) 비행 거리를 제외한 나머지에 대한 분석

```
df_numeric2 <- df_numeric[, -c(6)]

# 주성분 분석 수행
pca_result2 <- prcomp(df_numeric2, cor = TRUE)
## Warning: In prcomp.default(df_numeric2, cor = TRUE) :
##  extra argument 'cor' will be disregarded
```

```r
# 주성분(PC)의 설명력 확인
summary(pca_result2)
## Importance of components:
##                             PC1      PC2     PC3     PC4     PC5     PC6
     PC7
## Standard deviation     52.3666 15.14046 7.87492 2.52230 1.85743 1.8293
1.29960
## Proportion of Variance  0.8974  0.07502 0.02029 0.00208 0.00113 0.0011
0.00055
## Cumulative Proportion   0.8974  0.97244 0.99273 0.99481 0.99594 0.9970
0.99759
##                            PC8     PC9    PC10    PC11    PC12    PC13
    PC14
## Standard deviation     1.14500 1.06401 0.90388 0.85701 0.75320 0.71272
0.70318
## Proportion of Variance 0.00043 0.00037 0.00027 0.00024 0.00019 0.00017
0.00016
## Cumulative Proportion  0.99802 0.99839 0.99866 0.99890 0.99908 0.99925
0.99941
##                           PC15    PC16   PC17    PC18    PC19    PC20
   PC21
## Standard deviation     0.69022 0.63334 0.5662 0.49852 0.41542 0.35561
0.23422
## Proportion of Variance 0.00016 0.00013 0.0001 0.00008 0.00006 0.00004
0.00002
## Cumulative Proportion  0.99957 0.99970 0.9998 0.99988 0.99994 0.99998
1.00000
# 누적 설명력 계산 및 출력
cumulative_variance2 <- cumsum(pca_result2$sdev^2) /
sum(pca_result2$sdev^2) * 100
cat("\nCumulative variance explained:\n")
##
## Cumulative variance explained:
print(cumulative_variance2)
##  [1]  89.74187  97.24365  99.27310  99.48130  99.59421  99.70371  99.75
898
##  [8]  99.80189  99.83894  99.86567  99.88971  99.90827  99.92490  99.94
108
## [15]  99.95667  99.96980  99.98029  99.98842  99.99407  99.99820
100.00000
par(mfrow = c(1, 1))  # 다시 하나의 그래프 영역으로 복원

# ---------------------- Scree plot 그리기 ----------------------
plot(pca_result2, type = "l", main = "Scree Plot2")
```



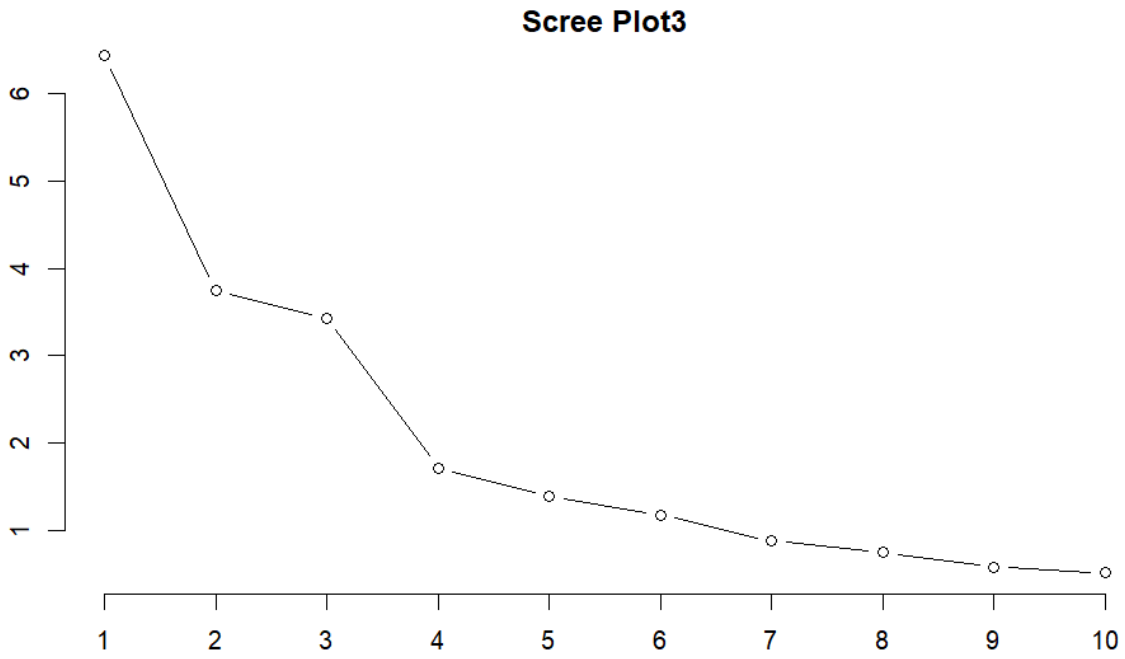

```r
# ---------------------- 주성분의 변수 기여도 ----------------------
# PC1 내의 변수 기여도(회전된 값의 제곱) 계산
pc2_contributions <- (pca_result2$rotation[, 1])^2

# 기여도가 높은 순서대로 상위 5 개 변수 선택
top_5_vars <- names(sort(pc2_contributions, decreasing = TRUE))[1:5]
pc2_contributions_top5 <- pc2_contributions[top_5_vars]

# 회전된 값 구하기
rotation_values <- pca_result2$rotation[, 1]

# 그래프 그리기
barplot_heights <- barplot(pc2_contributions_top5, col = "skyblue",
                           main = "Top 5 Variables in PC1 Contribution",
xlab = "Variables", ylab = "Contribution")

# 각 막대 내부에 기여도 값 추가
for (i in 1:length(barplot_heights)) {
  text(barplot_heights[i], pc2_contributions_top5[i] / 2, labels =
round(pc2_contributions_top5[i], 4), pos = 3, col = "black", cex = 0.8)
}

# 각 막대 위에 회전된 값 추가 (막대의 위로 올리기 위해 적절한 y 값 설정)
text(barplot_heights, pc2_contributions_top5 - 0.05 *
diff(range(pc2_contributions_top5)),
     labels = round(rotation_values[top_5_vars], 4), pos = 3, col =
"red", cex = 0.8)
```



2-3. PCA(3) 사용자 만족도(1~5)에 대한 분석

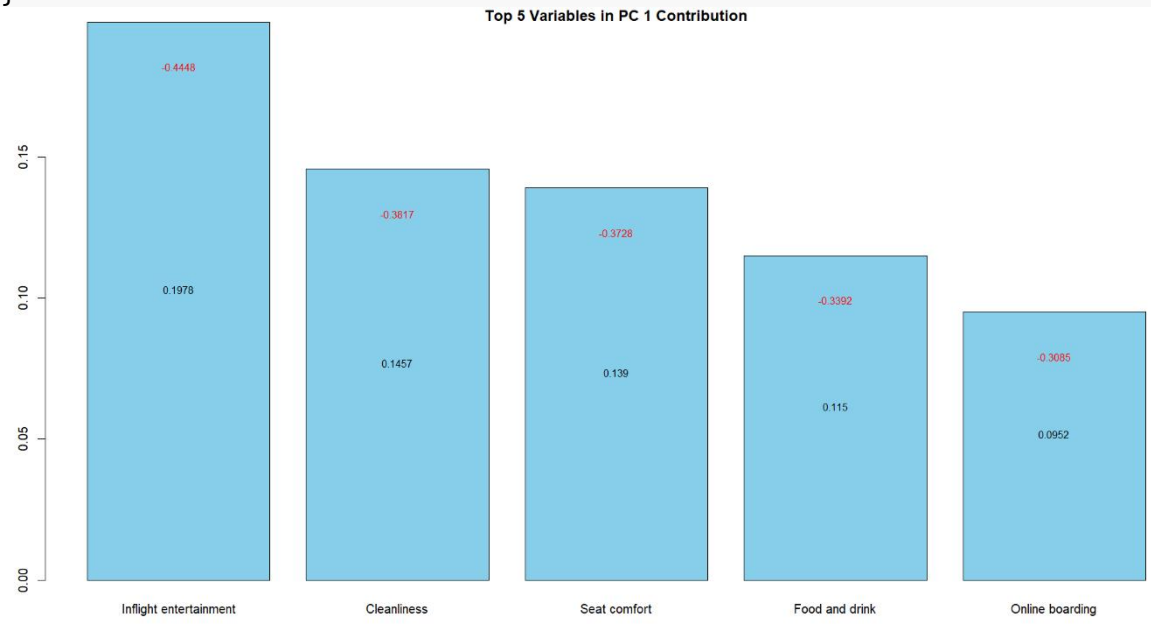
```
df_pca3 <- df_filled[, c(9:22)]
df_pca3 <- df_pca3[, -c(2)]

# 주성분 분석 수행
pca_result3 <- prcomp(df_pca3, cor = TRUE)
## Warning: In prcomp.default(df_pca3, cor = TRUE) :
##  extra argument 'cor' will be disregarded
# 주성분(PC)의 설명력 확인
summary(pca_result3)
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6
  PC7
## Standard deviation     2.5124 1.8574 1.8283 1.29707 1.14030 1.06498
0.90985
## Proportion of Variance 0.2987 0.1633 0.1582 0.07962 0.06153 0.05367
0.03918
## Cumulative Proportion  0.2987 0.4620 0.6202 0.69978 0.76131 0.81498
0.85416
##                            PC8     PC9    PC10    PC11    PC12    PC13

## Standard deviation     0.85717 0.74298 0.71415 0.70106 0.68371
0.57109
## Proportion of Variance 0.03477 0.02612 0.02414 0.02326 0.02212
0.01543
## Cumulative Proportion  0.88893 0.91505 0.93919 0.96244 0.98457
1.00000
# 누적 설명력 계산 및 출력
cumulative_variance3 <- cumsum(pca_result3$sdev^2) /
```

```r
sum(pca_result3$sdev^2) * 100
cat("\nCumulative variance explained:\n")
##
## Cumulative variance explained:
print(cumulative_variance3)
##  [1]  29.87202  46.19797  62.01596  69.97750  76.13080  81.49810  85.41
566
##  [8]  88.89269  91.50501  93.91854  96.24442  98.45659 100.00000
par(mfrow = c(1, 1))  # 다시 하나의 그래프 영역으로 복원


# ----------------------- Scree plot 그리기 -----------------------
plot(pca_result3, type = "l", main = "Scree Plot3")
```




```r
# ----------------------- 주성분의 변수 기여도 -----------------------
num_pcs <- 5  # 그래프를 그릴 주성분 개수
for (pc in 1:num_pcs) {
  # PC 의 기여도 계산
  pc_contributions <- (pca_result3$rotation[, pc])^2
  top_vars <- names(sort(pc_contributions, decreasing = TRUE))[1:5]  # 상위
5 개 변수 선택
  pc_contributions_top <- pc_contributions[top_vars]

  # PC 의 회전된 값 구하기
  rotation_values <- pca_result3$rotation[, pc]

  # 그래프 그리기
```

```r
  barplot_heights <- barplot(pc_contributions_top, col = "skyblue",
                             main = paste("Top 5 Variables in PC",pc,
"Contribution"),
                             xlab = "Variables", ylab = "Contribution")

  # 각 막대 내부에 기여도 값 추가
  for (i in 1:length(barplot_heights)) {
    text(barplot_heights[i], pc_contributions_top[i] / 2, labels =
round(pc_contributions_top[i], 4), pos = 3, col = "black", cex = 0.8)
  }

  # 각 막대 위에 회전된 값 추가 (막대의 위로 올리기 위해 적절한 y 값 설정)
  text(barplot_heights, pc_contributions_top-0.02,
       labels = round(rotation_values[top_vars], 4), pos = 3, col = "red",
cex = 0.8)

}
```



The end

1 차 주성분분석에서 비행거리가 압도적으로 많은 기여도를 가졌고, 이를 제외한 2 차 주성분분석에서는 도착 지연 시간과 출발 지연 시간이 많은 기여도를 차지했다. 1 에서 5 까지의 점수 형태를 가지는 만족도 점수 부분에 한정하여 3 차 주성분분석을 진행한 결과, 5 개의 주성분을 선택할 수 있었고 그 중 첫번째 주성분의 기여도를 살펴보았을 때 기내 여흥, 청결, 좌석의 편안함, 음식 그리고 온라인 탑승 수속에 대한 만족도가 차례대로 높은 설명력을 가진다.

```
Importance of components:
                          Comp.1    Comp.2    Comp.3     Comp.4    Comp.5     Comp.6
Standard deviation     2.0745097 1.4736404 1.3992760 1.37126399 1.2611358 1.02805957
Proportion of Variance 0.2265048 0.1142956 0.1030512 0.09896657 0.0837086 0.05562666
Cumulative Proportion  0.2265048 0.3408003 0.4438516 0.54281814 0.6265267 0.68215340
                           Comp.7     Comp.8     Comp.9    Comp.10    Comp.11    Comp.12
Standard deviation     0.98271555 0.92633955 0.83323942 0.82626057 0.69179820 0.68621955
Proportion of Variance 0.05082789 0.04516342 0.03654147 0.03593192 0.02518867 0.02478407
Cumulative Proportion  0.73298128 0.77814470 0.81468617 0.85061810 0.87580677 0.90059083
                          Comp.13    Comp.14    Comp.15    Comp.16    Comp.17    Comp.18
Standard deviation     0.62217279 0.60220729 0.59197688 0.53733595 0.51251599 0.43938988
Proportion of Variance 0.02037363 0.01908703 0.01844403 0.01519631 0.01382488 0.01016123
Cumulative Proportion  0.92096446 0.94005150 0.95849553 0.97369184 0.98751672 0.99767795
                          Comp.19
Standard deviation     0.210045041
Proportion of Variance 0.002322048
Cumulative Proportion  1.000000000
```

 # 상관행렬을 이용한 PCA

```
df <- df[, -c(1,2,10)]

pca_result <- princomp(df,cor=T)

pca_result

pca_result$loadings
```

표준화 변수를 이용한 PCA 결과 전체 변이의 73.30%를 설명하는 주성분 7개를 선택하였고, 첫 번째 주성분의 loading을 보았을 때 기내 만족도 변수 그룹에 속하는 레그룸 서비스의 계수가 가장 큰 절댓값을 나타낸다.

➔ **표준화 변수를 이용하지 않은 PCA의 3차 결과와 표준화 변수를 이용한 PCA 결과를 보면 기내 만족도가 항공사 만족에 주요하게 영향을 미치는 것을 알 수 있다.**

(2) LDA

```
set.seed(123)
index <- 1:nrow(data)
train_index <- sample(index, size = 0.7 * length(index))
test_index <- setdiff(index, train_index)
train_data <- data[train_index, -c(1,2,10)]
test_data <- data[test_index, -c(1,2,10)]

ld.result <- lda(satisfaction~.,data=train_data)
ld.result

## Call:
## lda(satisfaction ~ ., data = train_data)
##
## Prior probabilities of groups:
```

```
##         1         2
## 0.5567838 0.4432162
##
## Group means:
##     gender customerType      age typeOfTravel    class  distance     wifi
## 1 1.490419     1.755334 37.98667     1.491110 1.841169  935.3592
2.394292
## 2 1.493858     1.891922 41.60715     1.067999 1.282541 1524.6707
3.350971
##       eoob     gate foodAndDrink onlineBoarding seatComfort entertain
## 1 2.636585 2.990024     2.961814       2.726030    3.054326  2.881442
## 2 3.215096 2.945403     3.562337       4.121576    3.965753  3.962775
##   onboardService legRoomService baggageService checkinService
inflightService
## 1       3.011496       3.002037       3.364974       3.046128
3.386868
## 2       3.851967       3.830305       3.970964       3.656657
3.973198
##   cleanliness departDelay arrDelay
## 1    2.921897    15.79712 16.57497
## 2    3.753071    11.89056 11.98573
##
## Coefficients of linear discriminants:
##                          LD1
## gender           0.0538199808
## customerType     1.2317733323
## age             -0.0040466594
## typeOfTravel    -1.6630138502
## class           -0.3047603938
## distance         0.0000300205
## wifi             0.3688093945
## eoob            -0.0865101319
## gate            -0.1172792954
## foodAndDrink    -0.0231559191
## onlineBoarding   0.4003842290
## seatComfort     -0.0023439884
## entertain        0.0600998928
## onboardService   0.1285557462
## legRoomService   0.1346406914
## baggageService   0.0602854279
## checkinService   0.1295497776
## inflightService  0.0644882556
## cleanliness      0.1164562197
## departDelay      0.0011888120
## arrDelay        -0.0033828280

#neutral or dissatisfied = 1
#satisfied = 2


pc <- predict(ld.result,train_data)$class
correct.rate <- mean(train_data$satisfaction==pc)
error.rate <- mean(train_data$satisfaction!=pc)
```

```
correct.rate
```

```
## [1] 0.8787329
```

```
error.rate
```

```
## [1] 0.1212671
```

Train data 판별 결과 정분류율 87.87%, 오분류율 12.13%로 나타났다.

```
pc2 <- predict(ld.result,test_data)$class
correct.rate2 <- mean(test_data$satisfaction==pc2)
error.rate2 <- mean(test_data$satisfaction!=pc2)
correct.rate2
```

```
## [1] 0.8806621
```

```
error.rate2
```

```
## [1] 0.1193379
```

Test data 판별 결과 정분류율 88.07%, 오분류율 11.93%로 나타났다.

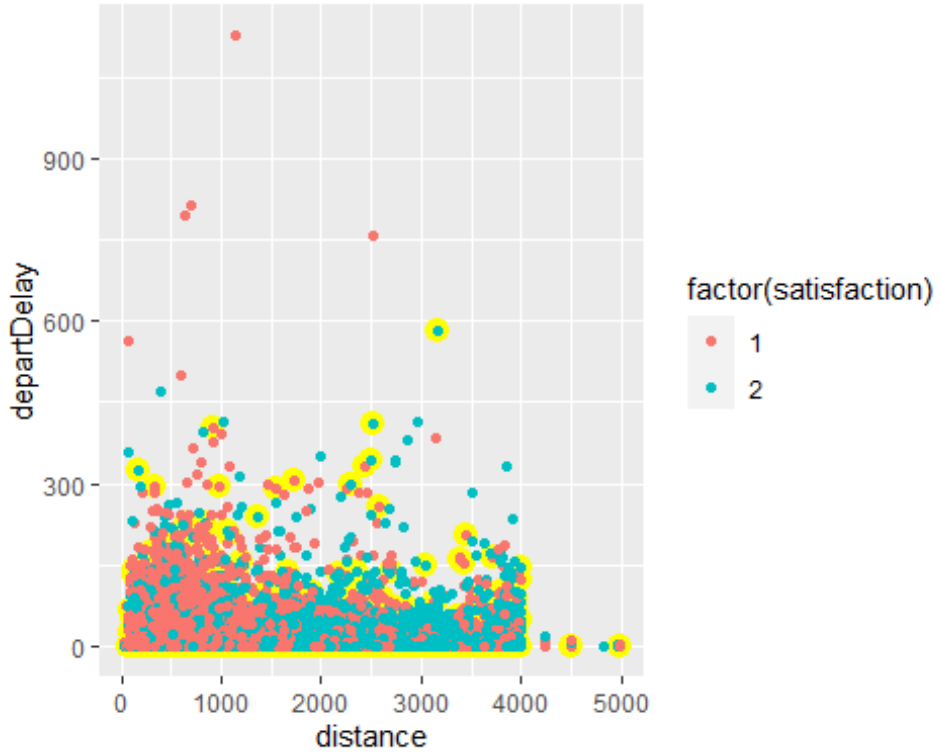
Train data 를 판별한 것보다 오분류율이 미세하게 낮아졌다.

```
train_data$pred <- pc
train_data$miss <- train_data$satisfaction!=pc
head(train_data)
```

```
##       gender customerType age typeOfTravel class distance wifi eoob gate
## 18847      2            2  29            1     1      347    5    5    5
## 18895      1            2  39            1     1     1690    2    4    2
## 25102      2            1  37            1     3      551    3    3    3
## 2986       1            2  31            2     2      224    4    4    4
## 1842       1            2  12            2     2      351    2    2    5
## 25718      2            2  31            1     2     1262    4    5    5
##       foodAndDrink onlineBoarding seatComfort entertain onboardService
## 18847            4              4           4         4              5
## 18895            5              4           4         5              5
## 25102            1              3           1         1              1
## 2986             5              4           5         5              2
## 1842             5              2           5         5              4
## 25718            4              4           4         4              2
##       legRoomService baggageService checkinService inflightService
cleanliness
## 18847              4              4              4               4
4
## 18895              5              5              4               5
```

```
5
## 25102              5              3              3              4
1
## 2986               2              3              2              3
5
## 1842               4              5              3              4
5
## 25718              2              4              4              3
4
##       departDelay arrDelay satisfaction pred  miss
## 18847          23       20            2    2 FALSE
## 18895           0        0            2    2 FALSE
## 25102           6        3            1    1 FALSE
## 2986            4        1            1    1 FALSE
## 1842            0        0            1    1 FALSE
## 25718           0        0            1    2  TRUE
```

```
ggplot(train_data,aes(distance,departDelay))+
  geom_point(data=train_data[train_data$miss,],col="yellow",size=4)+
  geom_point(aes(color=factor(satisfaction)))
```



비행 거리와 출발 지연 시간을 이용하여 산점도를 그려보았을 때 전체적으로 오분류된 점들이 보이지만, 특히 출발 시간이 지연되지 않은 경우에 오분류가 많이 일어나는 것을 확인할 수 있다.

(3) Clustering

최단연결법

```
x<-data[,c(3:9, 11:25)]
dx<-round(dist(x),digits=2)
#dx
D2<-dist(x,method="manhattan")
#D2
hc1<-hclust(dist(x)^2,method="single")

cut1 = cutree(hc1,k=4)
table(cut1)
```

```
## cut1
##     1     2     3     4
## 25972     1     2     1
```

최장연결법

```
hc2<-hclust(dist(x)^2,method="complete")

cut2 = cutree(hc2,k=4)
table(cut2)
```

```
## cut2
##     1     2     3     4
## 19611  2391  3954    20
```

평균연결법

```
hc3<-hclust(dist(x)^2,method="average")

cut3 = cutree(hc3, k=4)
table(cut3)
```

```
## cut3
##     1     2     3     4
## 17415  8548    11     2
```

K-means

```
data_k <-kmeans(x,centers=4)
table(data_k$cluster)
```

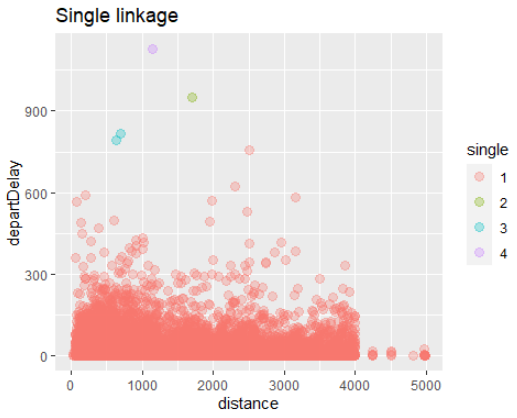
```
##
##     1     2     3     4
##  4495  2455  7097 11929
```

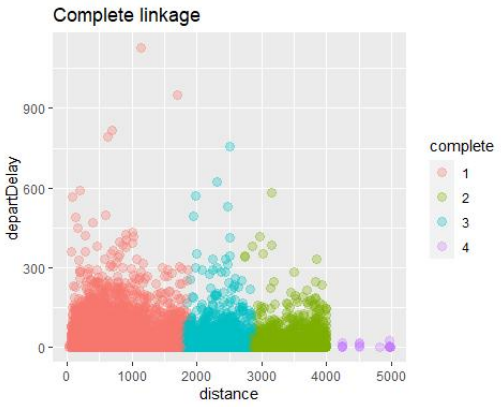
```
clus<-data.frame(data,kmeans=factor(data_k$cluster),
 single=factor(cutree(hc1,k=4)),
 complete=factor(cutree(hc2,k=4)),
 average=factor(cutree(hc3,k=4)))
```

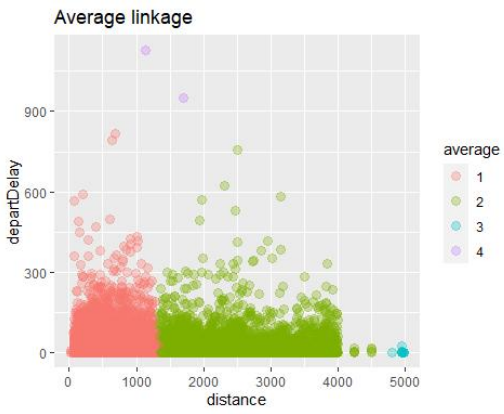
시각화 (X=distance, y=departDelay)

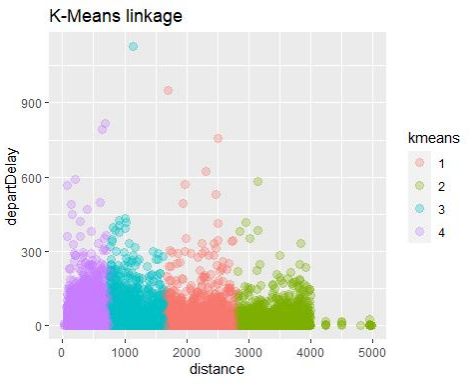
PCA 에서 비행거리와 도착지연시간이 항공사의 만족도에 크게 기여하였기 때문에, 비행거리와 도착지연시간의 그래프를 그려보았다.

```
ggplot(clus,aes(distance, departDelay))+
 geom_point(aes(color=single),size=3,alpha=0.3)+ggtitle("Single linkage")

ggplot(clus,aes(distance, departDelay))+
 geom_point(aes(color=complete),size=3,alpha=0.3)+ggtitle("Complete linkag
e")

ggplot(clus,aes(distance, departDelay))+
 geom_point(aes(color=average),size=3,alpha=0.3)+ggtitle("Average linkage
")

ggplot(clus,aes(distance, departDelay))+
 geom_point(aes(color=kmeans),size=3,alpha=0.3)+ggtitle("K-Means linkage")
```









k-means 를 제외한 군집이 잘 되지 않았음을 확인할 수 있었다. 최단연결법에서는 대부분의 데이터가 클러스터 1 에 속해있으며, 최장연결법에서는 클러스터 4 의 데이터는 매우 적은 것을 알 수 있다. 평균연결법에서도 또한 클러스터 3,4 의 데이터가 적었고, k-means 에서는 고르게 군집화됨을 볼 수 있습니다. 계층적 군집 방법에서 군집이 고르지 않아, 고르지 않게 하는 데이터를 제거하고 다시 클러스터링을 진행하였다.

```
cluster_data <- clus[!(clus$single %in% 2:4) & (clus$complete != 4) & !(clu
s$average %in% 3:4),]
```

최단연결법

```
x<-cluster_data[,c(3:9, 11:25)]
dx<-round(dist(x),digits=2)
#dx
D2<-dist(x,method="manhattan")
#D2
hc1<-hclust(dist(x)^2,method="single")

cut1 = cutree(hc1,k=4)
table(cut1)

## cut1
##     1     2     3     4
## 25949     1     1     1
```

최장연결법

```
hc2<-hclust(dist(x)^2,method="complete")

cut2 = cutree(hc2,k=4)
table(cut2)

## cut2
##     1     2     3     4
## 16187  2391  3420  3954
```

평균연결법

```
hc3<-hclust(dist(x)^2,method="average")

cut3 = cutree(hc3, k=4)
table(cut3)

## cut3
##     1     2     3     4
## 17413  6286  2246     7
```

K-Means

```
data_k <-kmeans(x,centers=4)
table(data_k$cluster)
```

```
##
##     1     2     3     4
##  4475  2456  7094 11927
```

(factor 로 변환)

```
clus_new<-data.frame(cluster_data[,c(3:9, 11:25)],kmeans=factor(data_k$clu
ster),
 single=factor(cutree(hc1,k=4)),
 complete=factor(cutree(hc2,k=4)),
 average=factor(cutree(hc3,k=4)))

str(clus_new)
```

```
## 'data.frame':    25952 obs. of  26 variables:
##  $ gender         : int  1 1 2 2 1 2 1 1 2 1 ...
##  $ customerType   : int  2 2 1 2 2 2 2 2 2 2 ...
##  $ age            : int  52 36 20 44 49 16 77 43 47 46 ...
##  $ typeOfTravel   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ class          : int  2 1 2 1 2 2 1 1 2 1 ...
##  $ distance       : int  160 2863 192 3377 1182 311 3987 2556 556 1744
 ...
##  $ wifi           : num  5 1 2 2.81 2 ...
##  $ eoob           : num  3 3 2 2.89 4 ...
##  $ gate           : int  4 1 4 2 3 3 5 2 2 2 ...
##  $ foodAndDrink   : num  3 5 2 3 4 5 3 4 5 3 ...
##  $ onlineBoarding : num  4 4 2 4 1 5 5 4 5 4 ...
##  $ seatComfort    : int  3 5 2 4 2 3 5 5 5 4 ...
##  $ entertain      : num  5 4 2 1 2 5 5 4 5 4 ...
##  $ onboardService : num  5 4 4 1 2 4 5 4 2 4 ...
##  $ legRoomService : num  5 4 1 1 2 3 5 4 2 4 ...
##  $ baggageService : int  5 4 3 1 2 1 5 4 5 4 ...
##  $ checkinService : int  2 3 2 3 4 1 4 5 3 5 ...
##  $ inflightService: num  5 4 2 1 2 2 5 4 3 4 ...
##  $ cleanliness    : num  5 5 2 4 4 5 3 3 5 4 ...
##  $ departDelay    : int  50 0 0 0 0 0 0 77 1 28 ...
##  $ arrDelay       : int  44 0 0 6 20 0 0 65 0 14 ...
##  $ satisfaction   : int  2 2 1 2 2 2 2 2 2 2 ...
##  $ kmeans         : Factor w/ 4 levels "1","2","3","4": 4 2 4 2 3 4 2 1 4
 1 ...
##  $ single         : Factor w/ 4 levels "1","2","3","4": 1 1 1 1 1 1 1 1 1
 1 ...
##  $ complete       : Factor w/ 4 levels "1","2","3","4": 1 2 1 2 3 1 2 4 1
 3 ...
##  $ average        : Factor w/ 4 levels "1","2","3","4": 1 2 1 3 1 1 3 2 1
 2 ...
```

```
clus_new$gender <- as.factor(clus_new$gender)
clus_new$customerType <- as.factor(clus_new$customerType)
clus_new$typeOfTravel <- as.factor(clus_new$typeOfTravel)
```

```
clus_new$class <- as.factor(clus_new$class)
clus_new$wifi <- as.factor(clus_new$wifi)
clus_new$eoob <- as.factor(clus_new$eoob)
clus_new$gate <- as.factor(clus_new$gate)
clus_new$foodAndDrink <- as.factor(clus_new$foodAndDrink)
clus_new$onlineBoarding <- as.factor(clus_new$onlineBoarding)
clus_new$seatComfort <- as.factor(clus_new$seatComfort)
clus_new$entertain <- as.factor(clus_new$entertain)
clus_new$onboardService <- as.factor(clus_new$onboardService)
clus_new$legRoomService <- as.factor(clus_new$legRoomService)
clus_new$baggageService <- as.factor(clus_new$baggageService)
clus_new$checkinService <- as.factor(clus_new$checkinService)
clus_new$inflightService <- as.factor(clus_new$inflightService)
clus_new$cleanliness <- as.factor(clus_new$cleanliness)
clus_new$satisfaction <- as.factor(clus_new$satisfaction)
```

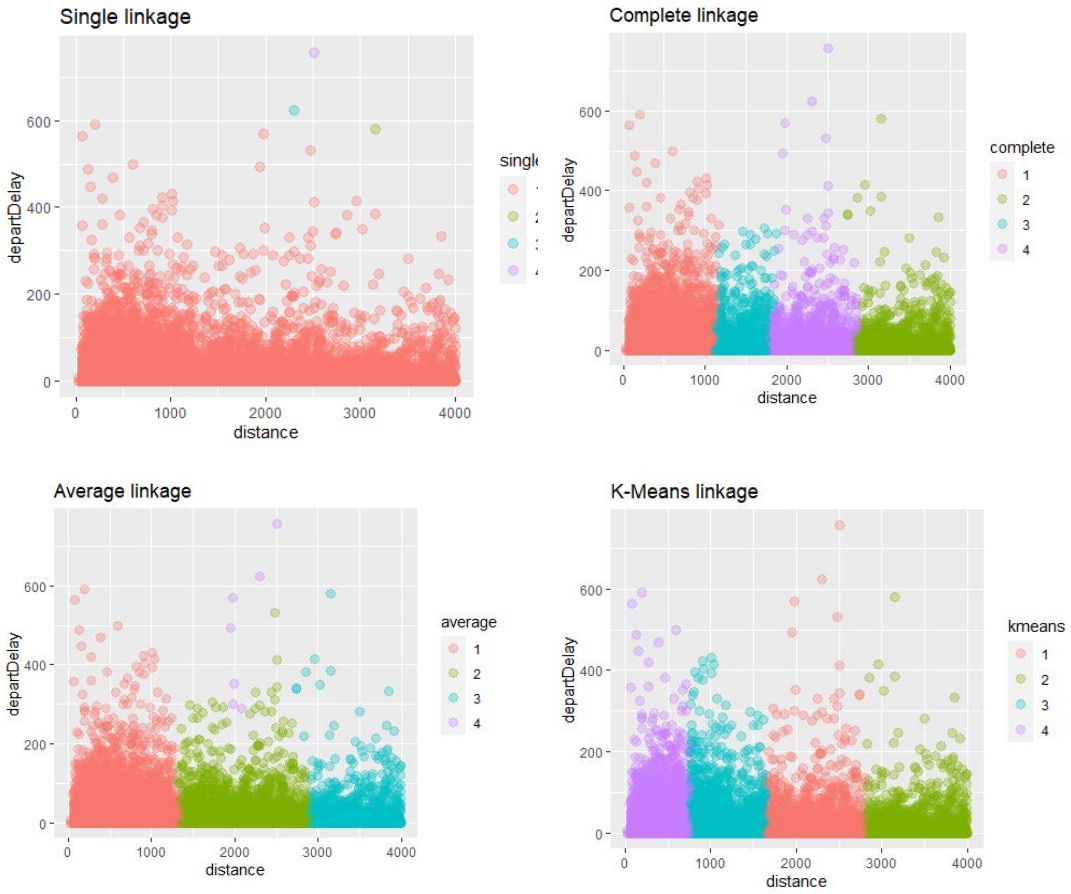
시각화 (X=distance, y=departDelay)

```
ggplot(clus_new,aes(distance, departDelay))+
 geom_point(aes(color=single),size=3,alpha=0.3)+ggtitle("Single linkage")
```

```
ggplot(clus_new,aes(distance, departDelay))+
 geom_point(aes(color=complete),size=3,alpha=0.3)+ggtitle("Complete linkag
e")
```

```
ggplot(clus_new,aes(distance, departDelay))+
 geom_point(aes(color=average),size=3,alpha=0.3)+ggtitle("Average linkage
")
```

```
ggplot(clus_new,aes(distance, departDelay))+
 geom_point(aes(color=kmeans),size=3,alpha=0.3)+ggtitle("K-Means linkage")
```



최장연결법과 k-means 가 치우침 없이 잘 군집화한 것을 확인할 수 있다. 클러스터링이 잘 된 최장연결법과 k-means 의 각 클러스터의 특징을 파악해보고자 한다.

군집별 시각화

1) 최장연결법

(i) 만족도

```
ggplot(clus_new,aes(satisfaction, fill=satisfaction))+
 geom_bar()+
  facet_wrap(~complete)
```

(ii) 비행거리

```
ggplot(clus_new,aes(complete, distance, color=complete))+
 geom_point()
```

(iii) 클래스

```
ggplot(clus_new,aes(class))+
 geom_bar(aes(fill=class))+
  facet_wrap(~complete)
```

(iv) 성별

```
ggplot(clus_new,aes(gender, fill=gender))+
 geom_bar()+
  facet_wrap(~complete)
```

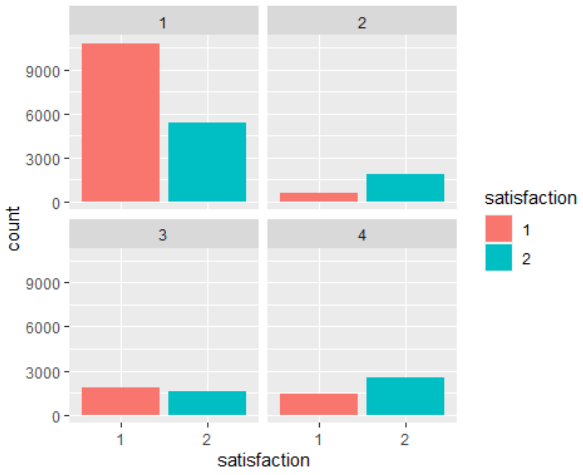
(v) 나이

```
ggplot(clus_new,aes(complete, age))+
  geom_boxplot(aes(fill=complete))
```

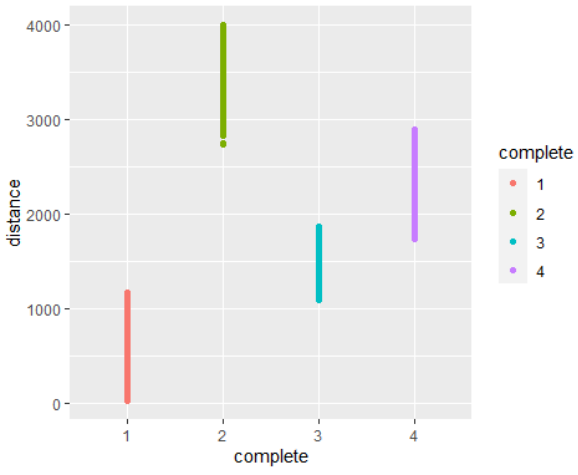
(vi) 도착지연시간

```
ggplot(clus_new,aes(complete, departDelay))+
  geom_point(aes(color=complete)) +
 geom_boxplot(aes(fill=complete))
```

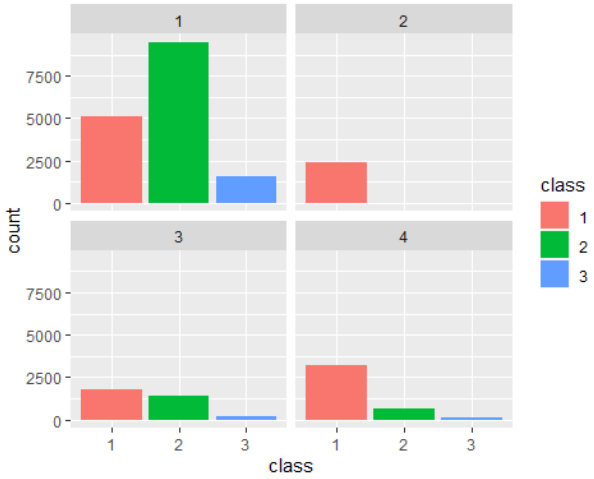
(i) 만족도 (1=불만족, 2 = 만족)



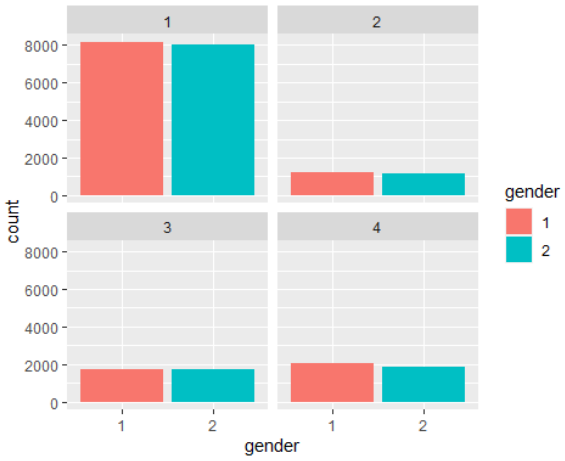
(ii) 비행거리



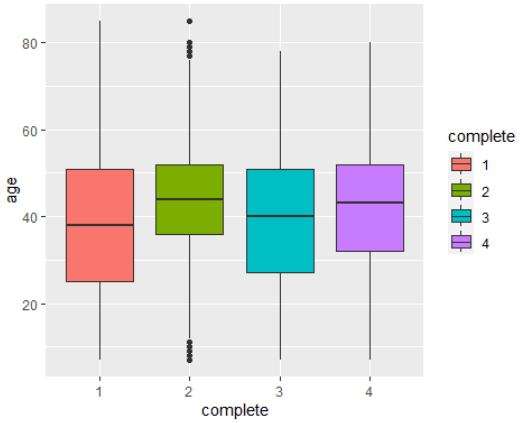
(iii) 클래스 (1 = 이코노미, 2 = 비즈니스, 3= 이코노미 플러스)



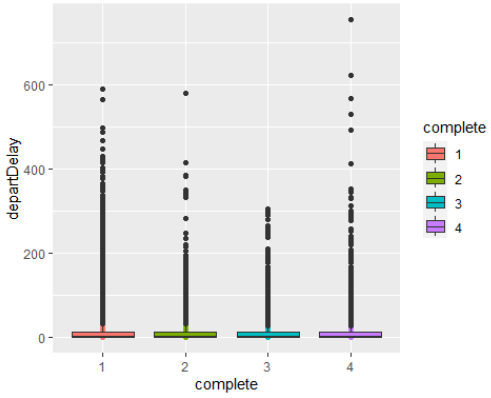
(iv) 성별 (1 = 여성, 2 = 남성)



(v) 나이



(vi) 비행거리



(i) 1번 군집은 불만족, 2,4번 군집은 불만족이 많으며, 3번 군집은 근소한 차이로 불만족이 많다.

(ii) 2, 4, 3, 1번 군집 순으로 비행거리가 짧아지는 것을 확인할 수 있다.

(iii) 1번 군집에서는 이코노미, 비즈니스, 이코노미 플러스 순으로 많지만, 다른 군집에서는 비즈니스, 이코노미, 이코노미 플러스 순으로 적어지는 것을 확인할 수 있다.

(iv) 고객 성별은 군집 별로 유의미한 차이를 보기 어렵다.

(v) 최장 연결법을 통해 군집1은 비교적 넓은 고객의 연령대를 포함하고 있으며, 군집 2는 약간 더 높은 나이 분포를 보인다. 군집 3과 4는 중간 나이대를 포함하며, 군집 2와 4는 비슷한 분포를 보인다.

(vi) 4번 군집이 도착지연시간이 길었을 때가 많았으며, 3번 군집은 도착지연시간이 다른 군집에 비해 짧다.

2) kmeans

(i) 만족도

```
ggplot(clus_new,aes(satisfaction, fill=satisfaction))+
 geom_bar()+
  facet_wrap(~kmeans)
```

(ii) 비행거리

```
ggplot(clus_new,aes(kmeans, distance, color=kmeans))+
 geom_point()
```

(iii) 클래스

```
ggplot(clus_new,aes(class))+
 geom_bar(aes(fill=class))+
  facet_wrap(~kmeans)
```

(iv) 성별

```
ggplot(clus_new,aes(gender, fill=gender))+
 geom_bar()+
  facet_wrap(~kmeans)
```

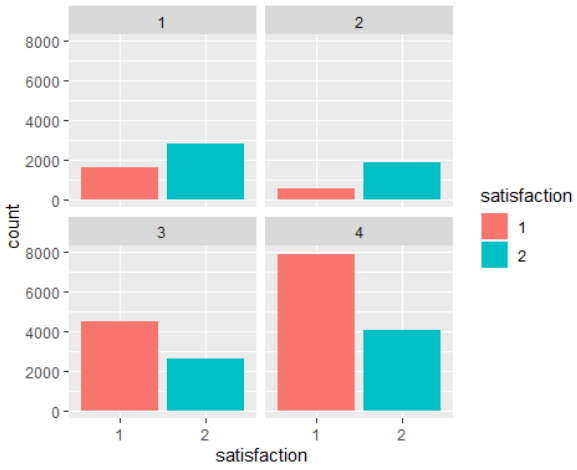
나이

```
ggplot(clus_new,aes(kmeans, age))+
  geom_boxplot(aes(fill=kmeans))
```

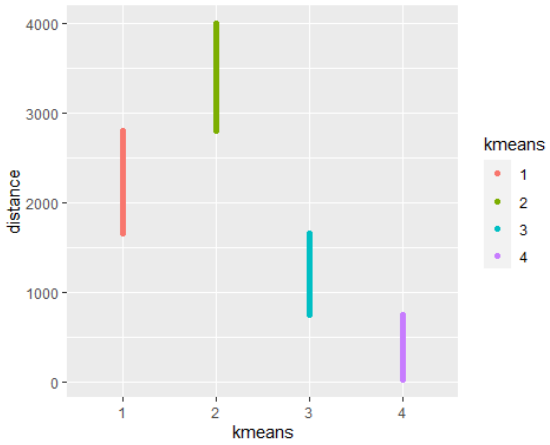
(vi) 도착지연시간

```
ggplot(clus_new,aes(kmeans, departDelay))+
  geom_point(aes(color=kmeans)) +
 geom_boxplot(aes(fill=kmeans))
```

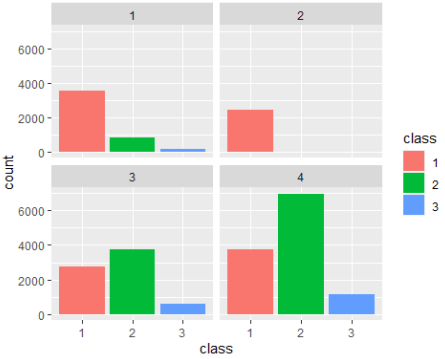
(i) 만족도 (1=불만족, 2 = 만족)



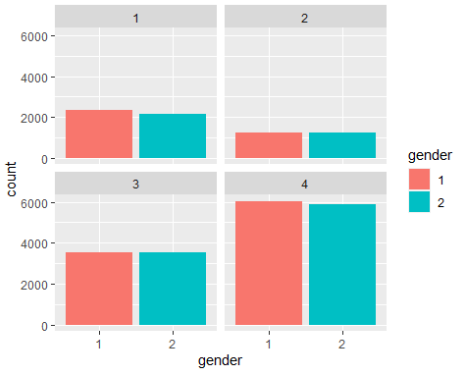
(ii) 비행거리



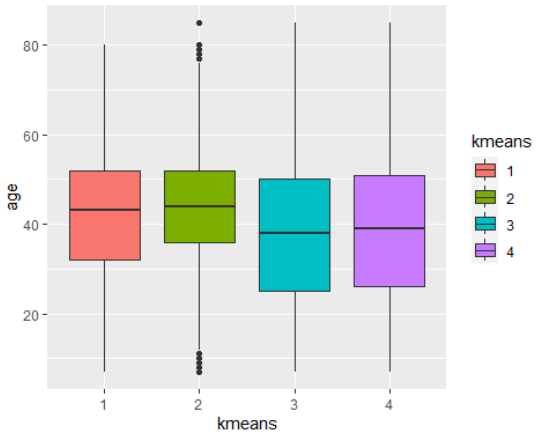
(iii) 클래스 (1 = 이코노미, 2 = 비즈니스, 3=이코노미 플러스)



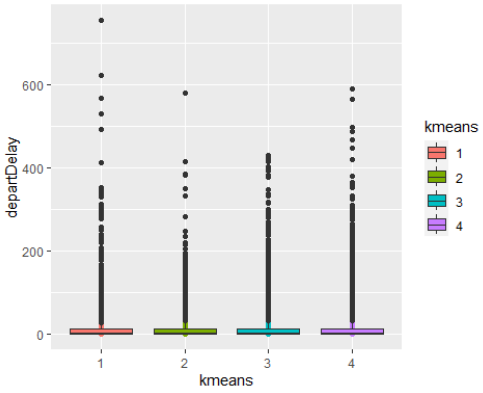
(iv) 성별 (1 = 여성, 2 = 남성)



(v) 나이



(vi) 비행거리



(i) 1,2번 군집은 만족, 3,4번 군집은 불만족이 많다.

(ii) 만족도가 만족이 많았던 1, 2번 군집을 비교하자면, 2번 군집은 1번 군집에 비해 비행거리가 길다. 만족도가 불만족이 많았던 3, 4번 군집을 비교하자면, 3번 군집에서는 4번 군집에 비해 비행거리가 길다.

(iii) (i)에서의 결과처럼 1,2 번 군집과 3,4 번 군집으로 나눌 수 있다. 1,2 번 군집은 비즈니스가 가장 많으며, 3,4 번 군집은 이코노미, 비즈니스, 이코노미플러스 순으로 많다.

(iv) 성별은 군집 별로 유의미한 차이를 보기 어렵다.

(v) k-means 클러스터링을 통해 나이의 분포가 군집에 따라 다르게 나타나는 것을 확인할 수 있다. 군집 1 은 상대적으로 고객의 나이가 많은 사람들이 많고, 군집 3 은 젊은 사람들이 많다. 군집 2 와 4 는 중간 나이대를 포함하고 있으며, 군집 2 는 일부 젊은 나이의 이상치를 포함하고 있다.

(vi) 4 번 군집이 도착지연시간이 길었던 때가 많았으며, 반대로 3 번 군집에서는 도착지연시간이 다른 군집보다 짧았던 때가 많다.

가장 잘 분류된 k-means 클러스터의 군집을 그래프를 통해 비교하자면, 군집 4 에서 고객 불만족 수가 가장 높았으며 만족과 불만족의 차이가 가장 컸다. 클러스터 4 에서 비행거리가 다른 클러스터보다 뚜렷하게 짧았다. 또한, 이코노비-비즈니스-이코노미 플러스 순으로 고객이 많았으며, 중간 연령층인 30 대 후반 고객이 많은 특징이 있음을 확인하였다. 도착지연시간이 긴 편에 속함을 알 수 있다.

4. 결론

(1) 표준화하지 않고 만족도 변수들만을 이용한 PCA 결과 첫번째 주성분의 주요 변수들이 기내 만족도 그룹에 속한 변수들이었고, (기내 여흥, 청결, 좌석의 편안함, 음식에 대한 만족도)

(2) 표준화 변수를 이용하여 PCA를 진행한 결과 첫번째 주성분의 주요 변수가 기내 만족도 그룹에 속하는 레그룸 서비스의 만족도였다.

두 결과를 종합하면 승객의 최종 항공사 만족도를 결정하는 주요 요인은 기내 만족도와 연관이 깊다고 판단할 수 있다.

Clustering 결과 비행 거리가 짧은 승객의 경우 항공사에 불만족하는 비율이 높았다. 이는 PCA에서 확인한 것과 같이 거리 변수의 단위가 1~5 사이에 존재하는 만족도 변수들과 매우 차이나기 때문에 도출된 결과일 가능성이 있으므로 해석에 주의해야 할 것으로 생각된다. 그러나 주어진 결과로만 해석하여 본다면, 항공사가 단거리 비행의 지연 시간을 단축하는 데에 힘쓸 경우 고객의 만족도를 향상시킬 수 있다고 보여진다.

항공사는 승객에게 제공되는 기내 서비스의 질과 단거리 비행의 지연 시간을 단축하도록 노력한다면 승객의 만족도 향상을 기대할 수 있을 것이다.

프로젝트 팀원 역할

신효진 – 피피티 제작, LDA 코드 작성, 표준화 변수를 이용한 PCA 코드 작성, 보고서 작성

이민채 – EDA 코드 작성, Clustering 코드 작성, 보고서 작성, 피피티 최종 정리, 보고서 작성

박경숙 – 데이터 전처리 코드 작성, EDA 코드 작성, PCA 코드 작성, 보고서 작성