

California Housing Prices

Sangkon Han(sangkon@pusan.ac.kr)

2023-04-05

Contents

캘리포니아 집 값(California Housing Prices)	1
데이터 불러오기	1
변수 요약 정보 확인	2
캘리포니아 집 값 예측 데이터 구조	3
시각화를 통한 데이터 확인	3
전처리	6
결측치(NA) 처리	6
후처리(Post-Cleaning)	7
불필요한 특징 삭제	8
범주형 변수(1)	8
수치형 변수 처리	9
정리된 데이터 결합	11
검증 데이터	11
예측 모델 생성 및 평가	12
단순 선형 모델	12
결론	13

캘리포니아 집 값(California Housing Prices)

해당 Competition의 데이터는 1990년 캘리포니아 인구조사 데이터인 캘리포니아 주택 가격(California Housing Prices) 데이터셋을 사용하며, 학습을 통해 주택 가격(median)을 예측하는 문제입니다.

데이터 불러오기

```
housing = read.csv("./data/housing.csv")
head(housing)
```

```
##   longitude latitude housing_median_age total_rooms total_bedrooms population
## 1   -122.23    37.88             41           880           129           322
## 2   -122.22    37.86             21          7099          1106          2401
## 3   -122.24    37.85             52          1467           190           496
## 4   -122.25    37.85             52          1274           235           558
## 5   -122.25    37.85             52          1627           280           565
## 6   -122.25    37.85             52           919           213           413
##   households median_income median_house_value ocean_proximity
## 1         126         8.3252         452600      NEAR BAY
## 2         1138         8.3014         358500      NEAR BAY
## 3          177         7.2574         352100      NEAR BAY
## 4          219         5.6431         341300      NEAR BAY
```

```
## 5      259      3.8462      342200      NEAR BAY
## 6      193      4.0368      269700      NEAR BAY
```

변수 요약 정보 확인

해당 데이터에 대한 정확한 정보를 모른다고 가정했을 때 가장 먼저 확인해야 하는 것은 데이터 자료형과 NA 값입니다.

```
str(housing)
```

```
## 'data.frame': 20640 obs. of 10 variables:
## $ longitude : num -122 -122 -122 -122 -122 ...
## $ latitude : num 37.9 37.9 37.9 37.9 37.9 ...
## $ housing_median_age: num 41 21 52 52 52 52 52 52 42 52 ...
## $ total_rooms : num 880 7099 1467 1274 1627 ...
## $ total_bedrooms : num 129 1106 190 235 280 ...
## $ population : num 322 2401 496 558 565 ...
## $ households : num 126 1138 177 219 259 ...
## $ median_income : num 8.33 8.3 7.26 5.64 3.85 ...
## $ median_house_value: num 452600 358500 352100 341300 342200 ...
## $ ocean_proximity : chr "NEAR BAY" "NEAR BAY" "NEAR BAY" "NEAR BAY" ...
```

기술 통계를 기반으로 한 정보도 함께 보도록 하겠습니다.

```
summary(housing)
```

```
## longitude latitude housing_median_age total_rooms
## Min. : -124.3 Min. : 32.54 Min. : 1.00 Min. : 2
## 1st Qu.: -121.8 1st Qu.: 33.93 1st Qu.: 18.00 1st Qu.: 1448
## Median : -118.5 Median : 34.26 Median : 29.00 Median : 2127
## Mean : -119.6 Mean : 35.63 Mean : 28.64 Mean : 2636
## 3rd Qu.: -118.0 3rd Qu.: 37.71 3rd Qu.: 37.00 3rd Qu.: 3148
## Max. : -114.3 Max. : 41.95 Max. : 52.00 Max. : 39320
##
## total_bedrooms population households median_income
## Min. : 1.0 Min. : 3 Min. : 1.0 Min. : 0.4999
## 1st Qu.: 296.0 1st Qu.: 787 1st Qu.: 280.0 1st Qu.: 2.5634
## Median : 435.0 Median : 1166 Median : 409.0 Median : 3.5348
## Mean : 537.9 Mean : 1425 Mean : 499.5 Mean : 3.8707
## 3rd Qu.: 647.0 3rd Qu.: 1725 3rd Qu.: 605.0 3rd Qu.: 4.7432
## Max. : 6445.0 Max. : 35682 Max. : 6082.0 Max. : 15.0001
## NA's : 207
## median_house_value ocean_proximity
## Min. : 14999 Length: 20640
## 1st Qu.: 119600 Class : character
## Median : 179700 Mode : character
## Mean : 206856
## 3rd Qu.: 264725
## Max. : 500001
##
```

기술 통계 정보를 기반으로 한 데이터를 통해서 확인할 수 있는 것은 아래와 같습니다.

1. total_bedrooms에 있는 207건의 결측값(NA)을 처리해야 합니다.
2. ocean_proximity는 binary column으로 변환해야 합니다.
3. 집 가격에 영향을 미치는 요소로 간주되는 total_bedrooms와 total_rooms등은 개별 가치로 파악할 수 있도록 mean_number_bedrooms 및 mean_number_rooms로 만들어야 됩니다.

캘리포니아 집 값 예측 데이터 구조

- longitude, 경도
- latitude, 위도
- housing_median_age, 주변의 집을 그룹화 했기 때문에 중앙값 사용
- total_rooms, 전체 방 수
- total_bedrooms, 전체 침실 수
- population, 인구
- households, 세대수
- median_income, 소득(중앙값)
- median_house_value, 주택 가격(중앙값)
- ocean_proximity, 해안 근접도

시각화를 통한 데이터 확인

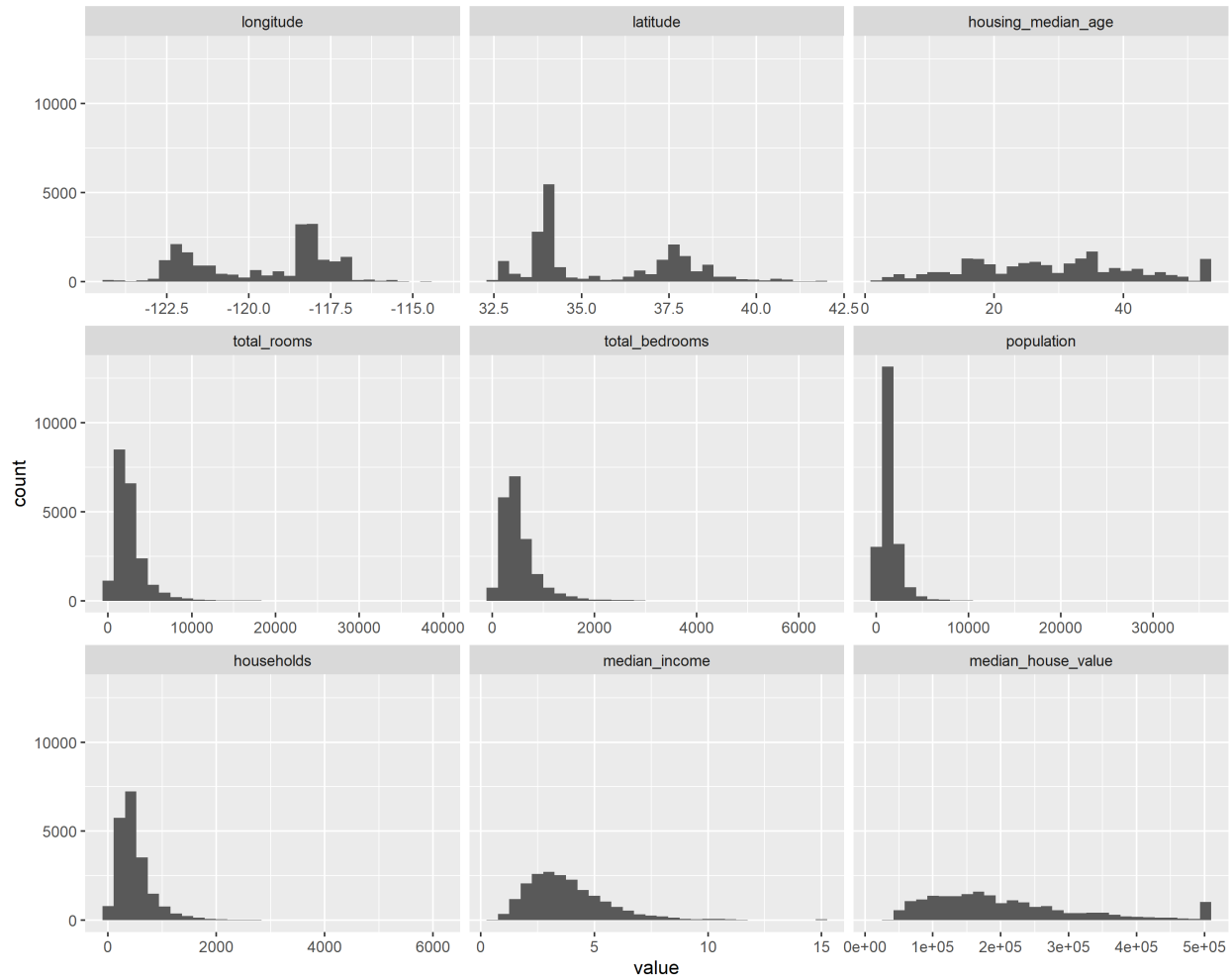
```
colnames(housing)
```

```
## [1] "longitude"      "latitude"        "housing_median_age"  
## [4] "total_rooms"    "total_bedrooms"  "population"  
## [7] "households"     "median_income"   "median_house_value"  
## [10] "ocean_proximity"
```

```
par(mfrow=c(2,5))
```

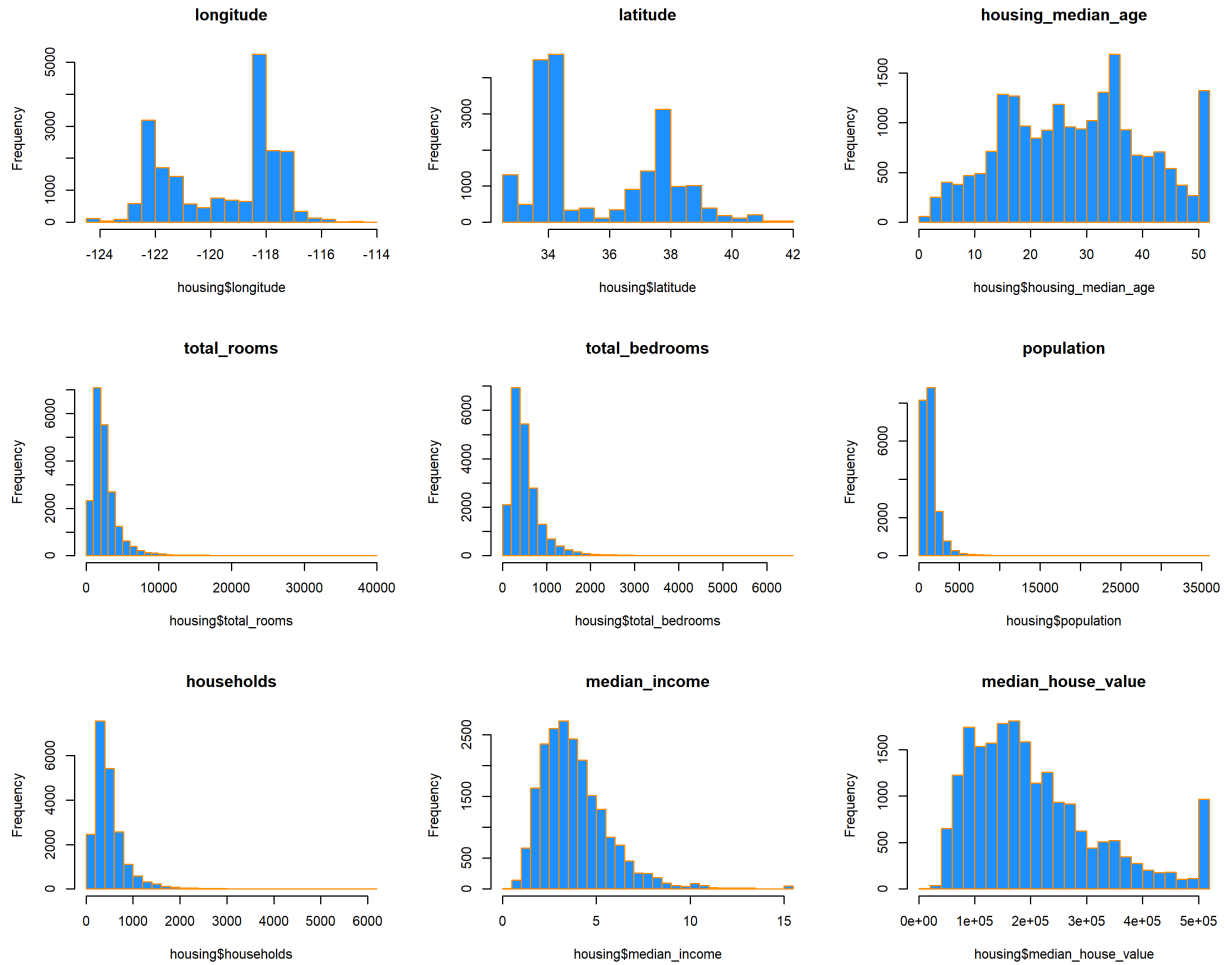
데이터 전체 분포를 확인하도록 하겠습니다.

```
ggplot(data = melt(housing), mapping = aes(x = value)) +  
  geom_histogram(bins = 30) +  
  facet_wrap(~variable, scales = 'free_x')
```

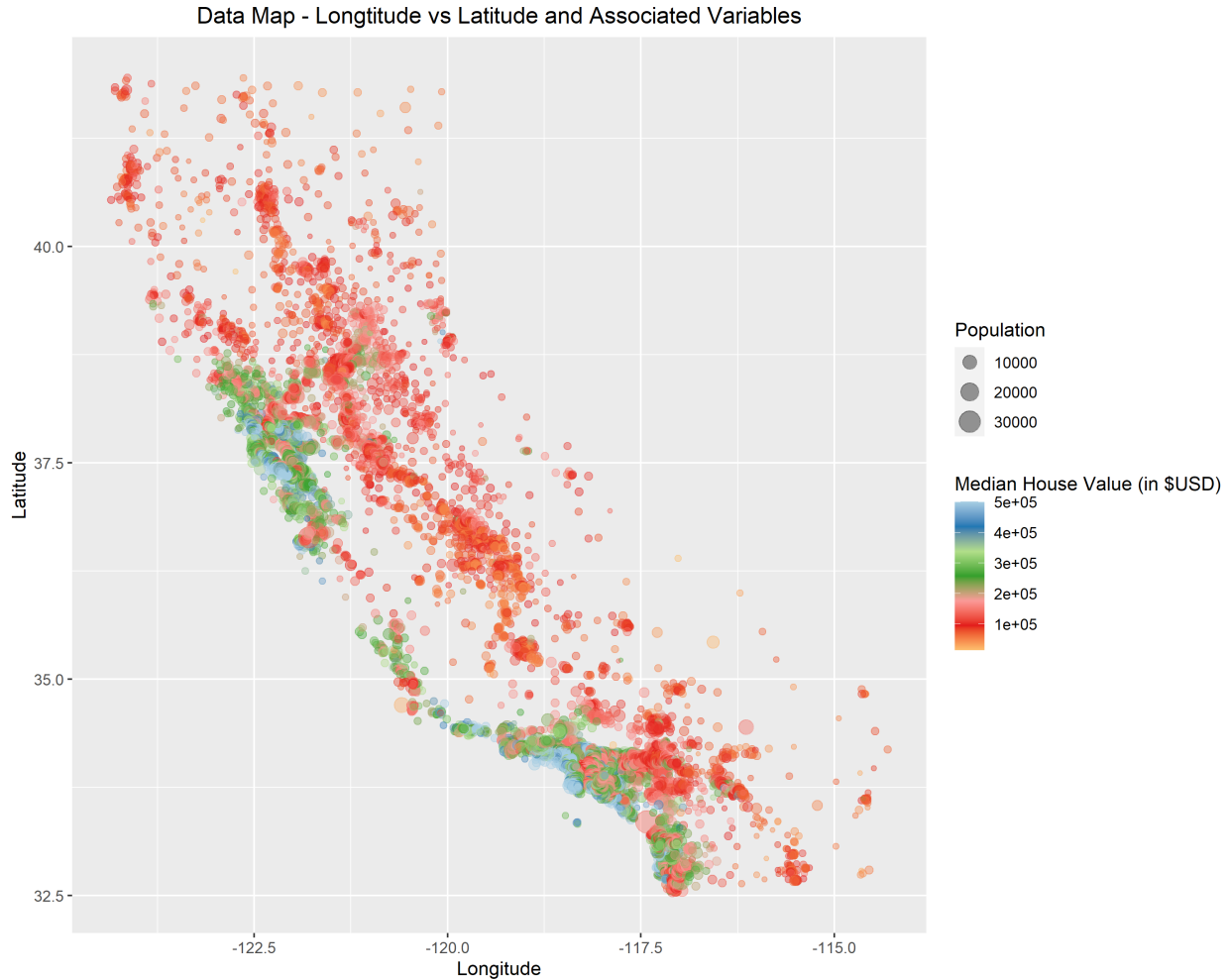


만약 ggplot등과 같은 형태의 데이터를 확인할 수 없다면, 아래와 같이 직접 히스토그램을 작성하셔도 됩니다.

```
par(mfrow = c(3, 3))
hist(housing$longitude, breaks = 30, main = "longitude", border="darkorange", col="dodgerblue")
hist(housing$latitude, breaks = 30, main = "latitude", border="darkorange", col="dodgerblue")
hist(housing$housing_median_age, breaks = 30, main = "housing_median_age", border="darkorange", col="dodgerblue")
hist(housing$total_rooms, breaks = 30, main = "total_rooms", border="darkorange", col="dodgerblue")
hist(housing$total_bedrooms, breaks = 30, main = "total_bedrooms", border="darkorange", col="dodgerblue")
hist(housing$population, breaks = 30, main = "population", border="darkorange", col="dodgerblue")
hist(housing$households, breaks = 30, main = "households", border="darkorange", col="dodgerblue")
hist(housing$median_income, breaks = 30, main = "median_income", border="darkorange", col="dodgerblue")
hist(housing$median_house_value, breaks = 30, main = "median_house_value", border="darkorange", col="dodgerblue")
```



```
plot_map <- ggplot(housing,
  aes(x = longitude, y = latitude, color = median_house_value,
      hma = housing_median_age, tr = total_rooms, tb = total_bedrooms,
      hh = households, mi = median_income)) +
  geom_point(aes(size = population), alpha = 0.4) +
  xlab("Longitude") +
  ylab("Latitude") +
  ggtitle("Data Map - Longitude vs Latitude and Associated Variables") +
  theme(plot.title = element_text(hjust = 0.5)) +
  scale_color_distiller(palette = "Paired") +
  labs(color = "Median House Value (in $USD)", size = "Population")
plot_map
```

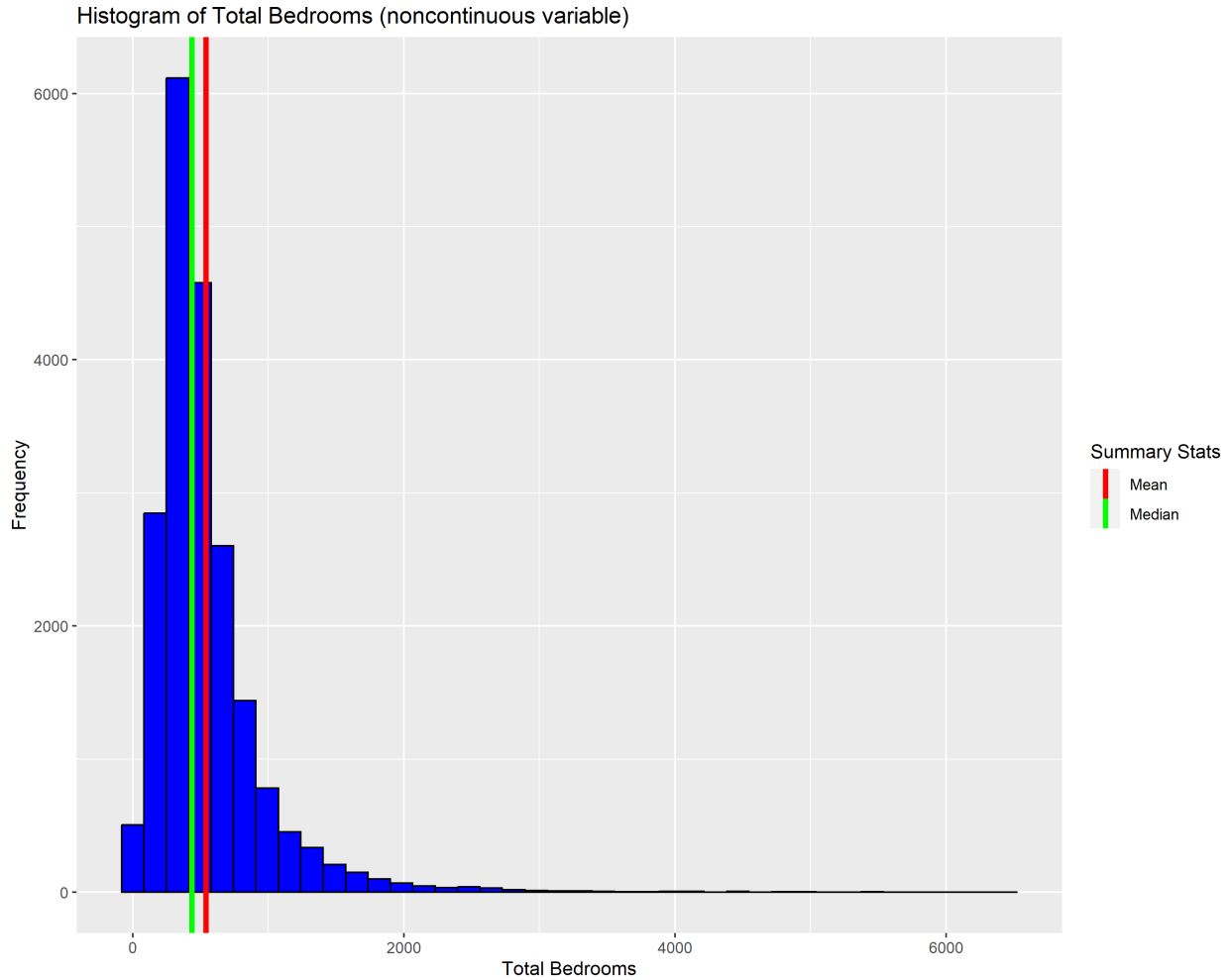


전처리

결측치(NA) 처리

평균에 비해서 극단값에 덜 민감한 중앙값을 사용해서 결측치를 해결하도록 하겠습니다. 그런데 가끔은 정말로 평균이 극단값에 덜 민감한지 확인하거나, 평균이나 중앙값 중 하나를 선택해야 할 때 어떤 것이 좋은 선택인지 궁금할 때가 있습니다. 잠시 확인해보고 넘어가보도록 하겠습니다.

```
bedroom_mean <- mean(housing$total_bedrooms, na.rm=TRUE)
bedroom_median <- median(housing$total_bedrooms, na.rm=TRUE)
ggplot(housing, aes(x = total_bedrooms)) +
  geom_histogram(bins = 40, color = "black", fill = "blue") +
  geom_vline(aes(xintercept = bedroom_mean, color = "Mean"), lwd = 1.5) +
  geom_vline(aes(xintercept = bedroom_median, color = "Median"), lwd = 1.5) +
  xlab("Total Bedrooms") +
  ylab("Frequency") +
  ggtitle("Histogram of Total Bedrooms (noncontinuous variable)") +
  scale_color_manual(name = "Summary Stats", labels = c("Mean", "Median"), values = c("red", "green"))
```



히스토그램에서 데이터 분포를 살펴보면 `total_bedrooms` 변수의 중앙값을 사용하는 것이 더 좋을 듯 합니다. 선택은 각자의 몫이지만, 해당 히스토그램을 통해서 결정에 대한 근거를 찾을 수 있습니다. NA값을 중앙값으로 처리하도록 하겠습니다.

```
housing$total_bedrooms[is.na(housing$total_bedrooms)] <- median(housing$total_bedrooms, na.rm=TRUE)
sum(is.na(housing))
```

```
## [1] 0
```

후처리(Post-Cleaning)

데이터를 정리한 후 데이터셋의 구조를 보면 factor 변수인 `ocean_proximity` 외에도 9개의 numeric 변수가 있음을 알 수 있습니다. 이 중 3개는 연속적(continuous)(`longitude`, `latitude`, `median_income`)이고 6개는 불연속적(discrete)(`housing_median_age`, `total_rooms`, `total_bedrooms`, `population`, `households`, `median_house_value`) 입니다..

```
housing$mean_bedrooms <- housing$total_bedrooms/housing$households
housing$mean_rooms <- housing$total_rooms/housing$households
head(housing)
```

```
##   longitude latitude housing_median_age total_rooms total_bedrooms population
## 1   -122.23    37.88             41         880         129         322
## 2   -122.22    37.86             21        7099        1106        2401
## 3   -122.24    37.85             52        1467         190         496
```

```
## 4 -122.25 37.85 52 1274 235 558
## 5 -122.25 37.85 52 1627 280 565
## 6 -122.25 37.85 52 919 213 413
## households median_income median_house_value ocean_proximity mean_bedrooms
## 1 126 8.3252 452600 NEAR BAY 1.0238095
## 2 1138 8.3014 358500 NEAR BAY 0.9718805
## 3 177 7.2574 352100 NEAR BAY 1.0734463
## 4 219 5.6431 341300 NEAR BAY 1.0730594
## 5 259 3.8462 342200 NEAR BAY 1.0810811
## 6 193 4.0368 269700 NEAR BAY 1.1036269
## mean_rooms
## 1 6.984127
## 2 6.238137
## 3 8.288136
## 4 5.817352
## 5 6.281853
## 6 4.761658
```

불필요한 특징 삭제

머신러닝의 학습에 사용되지 않을 불필요한 특징은 삭제하도록 하겠습니다.

```
drops <- c('total_bedrooms', 'total_rooms')
housing <- housing[, !(names(housing) %in% drops)]
head(housing)
```

```
## longitude latitude housing_median_age population households median_income
## 1 -122.23 37.88 41 322 126 8.3252
## 2 -122.22 37.86 21 2401 1138 8.3014
## 3 -122.24 37.85 52 496 177 7.2574
## 4 -122.25 37.85 52 558 219 5.6431
## 5 -122.25 37.85 52 565 259 3.8462
## 6 -122.25 37.85 52 413 193 4.0368
## median_house_value ocean_proximity mean_bedrooms mean_rooms
## 1 452600 NEAR BAY 1.0238095 6.984127
## 2 358500 NEAR BAY 0.9718805 6.238137
## 3 352100 NEAR BAY 1.0734463 8.288136
## 4 341300 NEAR BAY 1.0730594 5.817352
## 5 342200 NEAR BAY 1.0810811 6.281853
## 6 269700 NEAR BAY 1.1036269 4.761658
```

범주형 변수(1)

범주형 변수 처리를 위해서 별도의 데이터 프레임을 생성합니다.

```
categories <- unique(housing$ocean_proximity)
cat_housing <- data.frame(ocean_proximity = housing$ocean_proximity)
head(cat_housing)
```

```
## ocean_proximity
## 1 NEAR BAY
## 2 NEAR BAY
## 3 NEAR BAY
## 4 NEAR BAY
## 5 NEAR BAY
## 6 NEAR BAY
```


모든 값이 0으로 채워진 데이터 프레임 생성합니다.

```
for(cat in categories){
  cat_housing[,cat] = rep(0, times= nrow(cat_housing))
}
```

```
head(cat_housing)
```

```
##   ocean_proximity NEAR BAY <1H OCEAN INLAND NEAR OCEAN ISLAND
## 1      NEAR BAY      0      0      0      0      0
## 2      NEAR BAY      0      0      0      0      0
## 3      NEAR BAY      0      0      0      0      0
## 4      NEAR BAY      0      0      0      0      0
## 5      NEAR BAY      0      0      0      0      0
## 6      NEAR BAY      0      0      0      0      0
```

필요한 데이터만 1로 업데이트 합니다.

```
for(i in 1:length(cat_housing$ocean_proximity)){
  cat <- as.character(cat_housing$ocean_proximity[i])
  cat_housing[,cat][i] <- 1
}
```

```
head(cat_housing)
```

```
##   ocean_proximity NEAR BAY <1H OCEAN INLAND NEAR OCEAN ISLAND
## 1      NEAR BAY      1      0      0      0      0
## 2      NEAR BAY      1      0      0      0      0
## 3      NEAR BAY      1      0      0      0      0
## 4      NEAR BAY      1      0      0      0      0
## 5      NEAR BAY      1      0      0      0      0
## 6      NEAR BAY      1      0      0      0      0
```

기존 특징은 사용하지 않기 때문에 삭제합니다.

```
cat_columns <- names(cat_housing)
keep_columns <- cat_columns[cat_columns != 'ocean_proximity']
cat_housing <- select(cat_housing, one_of(keep_columns))
tail(cat_housing)
```

```
##      NEAR BAY <1H OCEAN INLAND NEAR OCEAN ISLAND
## 20635      0      0      1      0      0
## 20636      0      0      1      0      0
## 20637      0      0      1      0      0
## 20638      0      0      1      0      0
## 20639      0      0      1      0      0
## 20640      0      0      1      0      0
```

수치형 변수 처리

수치의 단위(unit)이 일정하지 않기 때문에 수치형 변수를 일괄로 처리하도록 하겠습니다. 먼저 특징을 확인합니다.

```
colnames(housing)
```

```
## [1] "longitude"      "latitude"        "housing_median_age"
## [4] "population"     "households"      "median_income"
## [7] "median_house_value" "ocean_proximity" "mean_bedrooms"
```

```
## [10] "mean_rooms"
```

명목형 변수(ocean_proximity)와 예측 변수(median_house_value)는 대상에서 제외하도록 하겠습니다.

```
drops <- c('ocean_proximity', 'median_house_value')
housing_num <- housing[, !(names(housing) %in% drops)]
head(housing_num)
```

```
##      longitude latitude housing_median_age population households median_income
## 1    -122.23    37.88             41          322         126         8.3252
## 2    -122.22    37.86             21         2401        1138         8.3014
## 3    -122.24    37.85             52          496         177         7.2574
## 4    -122.25    37.85             52          558         219         5.6431
## 5    -122.25    37.85             52          565         259         3.8462
## 6    -122.25    37.85             52          413         193         4.0368
##      mean_bedrooms mean_rooms
## 1      1.0238095    6.984127
## 2      0.9718805    6.238137
## 3      1.0734463    8.288136
## 4      1.0730594    5.817352
## 5      1.0810811    6.281853
## 6      1.1036269    4.761658
```

대부분의 데이터는 자신만의 단위(Unit)을 사용합니다. 예를 들어, 한국에서 사용하는 대표적인 단위가 ‘평수’입니다. ‘년’, ‘km’ 등 과 같은 표준적인 단위도 있지만, ‘마일’, ‘피트’와 같은 특정 문화권에서 사용하는 단위도 있습니다. 그리고 단위에 따른 값의 범위도 꽤 차이가 있습니다. 단위가 다르면 직접적인 비교가 불가능합니다. 그래서 일반적으로 데이터를 정규화 또는 표준화를 진행합니다. 문제는 정규화와 표준화에 대한 이해도가 생각보다 낮다는 점입니다.

정규화(normalization)는 특성 내에 가장 큰 값은 1로, 가장 작은 값은 0으로 변환합니다. 공식은 아래와 같습니다.

$$\frac{x - x_{min}}{x_{max} - x_{min}}$$

표준화(standardization)는 어떤 특성의 값들이 정규분포, 즉 종모양의 분포를 따른다고 가정하고 값들을 0의 평균, 1의 표준편차를 갖도록 변환해주는 것입니다. 공식은 아래와 같습니다.

$$\frac{x - \mu}{\sigma} (\mu : \text{평균}, \sigma : \text{표준편차})$$

해당 데이터는 정규화가 아니라 ‘표준화’를 사용합니다.

```
scaled_housing_num <- scale(housing_num)
head(scaled_housing_num)
```

```
##      longitude latitude housing_median_age population households median_income
## [1,] -1.327803  1.052523      0.9821189 -0.9744050 -0.9770092    2.34470896
## [2,] -1.322812  1.043159     -0.6070042  0.8614180  1.6699206    2.33218146
## [3,] -1.332794  1.038478      1.8561366 -0.8207575 -0.8436165    1.78265622
## [4,] -1.337785  1.038478      1.8561366 -0.7660095 -0.7337637    0.93294491
## [5,] -1.337785  1.038478      1.8561366 -0.7598283 -0.6291419   -0.01288068
## [6,] -1.337785  1.038478      1.8561366 -0.8940491 -0.8017678    0.08744452
##      mean_bedrooms mean_rooms
## [1,] -0.148510661  0.6285442
## [2,] -0.248535936  0.3270334
```

```
## [3,] -0.052900657 1.1555925
## [4,] -0.053646030 0.1569623
## [5,] -0.038194658 0.3447024
## [6,] 0.005232996 -0.2697231
```

정리된 데이터 결합

```
cleaned_housing <- cbind(cat_housing, scaled_housing_num, median_house_value=housing$median_house_value)
head(cleaned_housing)
```

```
##      NEAR BAY <1H OCEAN INLAND NEAR OCEAN ISLAND longitude latitude
## 1         1         0         0         0         0 -1.327803 1.052523
## 2         1         0         0         0         0 -1.322812 1.043159
## 3         1         0         0         0         0 -1.332794 1.038478
## 4         1         0         0         0         0 -1.337785 1.038478
## 5         1         0         0         0         0 -1.337785 1.038478
## 6         1         0         0         0         0 -1.337785 1.038478
##      housing_median_age population households median_income mean_bedrooms
## 1          0.9821189 -0.9744050 -0.9770092      2.34470896 -0.148510661
## 2         -0.6070042  0.8614180  1.6699206      2.33218146 -0.248535936
## 3          1.8561366 -0.8207575 -0.8436165      1.78265622 -0.052900657
## 4          1.8561366 -0.7660095 -0.7337637      0.93294491 -0.053646030
## 5          1.8561366 -0.7598283 -0.6291419     -0.01288068 -0.038194658
## 6          1.8561366 -0.8940491 -0.8017678      0.08744452  0.005232996
##      mean_rooms median_house_value
## 1  0.6285442          452600
## 2  0.3270334          358500
## 3  1.1555925          352100
## 4  0.1569623          341300
## 5  0.3447024          342200
## 6 -0.2697231          269700
```

데이터 전/후 처리가 완료되었습니다. 이제 머신러닝을 진행해보도록 하겠습니다.

검증 데이터

이번 단계에서는 전체 데이터에서 학습 데이터(train)와 검증 데이터(test)를 분리합니다. 검증 데이터는 학습된 모델의 평가에만 사용되며, 학습/검증 데이터 분리를 통해 예측 결과의 객관성을 확보할 수 있습니다.

```
set.seed(42)
sample <- sample.int(n = nrow(cleaned_housing), size = floor(.8*nrow(cleaned_housing)), replace = F)
train <- cleaned_housing[sample, ] #just the samples
test  <- cleaned_housing[-sample, ] #everything but the samples
head(train)
```

```
##      NEAR BAY <1H OCEAN INLAND NEAR OCEAN ISLAND longitude latitude
## 18753         0         0         1         0         0 -1.4226356 2.3400047
## 9290         1         0         0         0         0 -1.4675563 1.1367943
## 1252         0         0         1         0         0 -1.2978559 1.7220134
## 15506         0         0         0         1         0  1.1528165 -1.1947183
## 8826         0         1         0         0         0  0.6037860 -0.7218613
## 10289         0         1         0         0         0  0.8633277 -0.8154964
##      housing_median_age population households median_income mean_bedrooms
## 18753         -0.4480919 -0.25914928 -0.14526644      -1.1726211  0.0008497704
## 9290         -0.3686357  3.29946864  3.61065366      -0.4192346 -0.1380210149
```

```
## 1252      0.5053819 -0.02691193 -0.06156905    -0.6885757  0.1131708564
## 15506     -1.3221096  0.47023494  0.56616133     0.7999851  0.0692684052
## 8826      0.3464696  0.37045235  1.65945840    -0.9738656 -0.1332234247
## 10289     -0.7659165  0.31923651  0.15813658     1.5333170 -0.2562856779
##      mean_rooms median_house_value
## 18753 -0.1479026          80400
## 9290  -0.1533974          118800
## 1252   0.1755923           62700
## 15506  0.6733451          299600
## 8826  -0.9600007          243800
## 10289  0.3635848          264300
```

분리된 데이터가 전체 데이터를 반영하고 있는지 확인합니다.

```
nrow(train) + nrow(test) == nrow(cleaned_housing)
```

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

예측 모델 생성 및 평가

단순 선형 모델

간단한 선형 모델 테스트를 위해 아래 3개 변수를 선택하여 분석에 적용합니다. - 소득(중앙값) : median_income
- 방 수(평균값) : mean_rooms - 인구 : population

또한, 모델의 과적합(overfit) 문제를 피하기 위해 cv.glm함수를 이용하여 교차 검증(k_fold)를 수행하며, 여기서는 모델 테스트에 전처리된 데이터 자체를 사용합니다.

```
glm_house = glm(median_house_value~median_income+mean_rooms+population, data=cleaned_housing)
k_fold_cv_error = cv.glm(cleaned_housing , glm_house, K=5)
k_fold_cv_error$delta
```

```
## [1] 6936197680 6933908813
```

첫 번째 성분은 예측 오차의 원시 교차 검증 추정치입니다. 두 번째 구성 요소는 조정된 교차 검증 추정치입니다.

RMSE(평균 제곱근 오차) 값을 출력합니다. RMSE는 회귀 예측 모델에 대한 두 개의 주요 성과 지표 중 하나입니다. 평균 제곱근 오차는 예측 모델에서 예측한 값과 실제 값 사이의 평균 차이를 측정합니다. 예측 모델이 목표 값(정확도)을 얼마나 잘 예측할 수 있는지 추정합니다. 일반적으로 회귀 모델은 MAE(평균 절대 오차)가 적당하지만, 이상치에 민감한 단점이 있습니다. 해당 데이터는 기본적으로 이상치가 많이 포함된 데이터이기 때문에 MAE가 아닌 RMSE를 사용하도록 하였습니다.

```
glm_cv_rmse = sqrt(k_fold_cv_error$delta)[1]
glm_cv_rmse
```

```
## [1] 83283.84
```

glm에서 제공하는 다양한 정보는 아래에서 확인할 수 있습니다.

```
names(glm_house)
```

```
## [1] "coefficients"      "residuals"        "fitted.values"
## [4] "effects"          "R"                 "rank"
## [7] "qr"               "family"            "linear.predictors"
## [10] "deviance"         "aic"               "null.deviance"
## [13] "iter"             "weights"           "prior.weights"
## [16] "df.residual"      "df.null"           "y"
## [19] "converged"        "boundary"          "model"
## [22] "call"             "formula"           "terms"
## [25] "data"             "offset"            "control"
```

```
## [28] "method"          "contrasts"        "xlevels"
```

이중에서 가장 중요한 정보를 제공하는 `coefficients`를 확인하도록 하겠습니다.

```
glm_house$coefficients
```

```
##      (Intercept) median_income    mean_rooms    population  
##      206855.817      82608.959      -9755.442      -3948.293
```

```
## (Intercept) median_income    mean_rooms    population  
## 206855.817      82608.959      -9755.442      -3948.293
```

결론

분석을 통해 소득 중앙값(`median_income`)이 주택 가격(`median_house_value`)에 가장 큰 영향을 미친다고 판단할 수 있습니다.