Linear Regression in R

Gladys Dalla

2025-03-11

Boston Dataset Analysis

Objective

The objective of this analysis is to predict median housing values (`medv`) in Boston suburbs using:

1. A simple linear regression model with one predictor: percentage of lower status population (`lstat`)

2. A multiple regression model with two predictors: `lstat` and proportion of owner-occupied units built prior to 1940 (`age`)

This analysis will help us understand how socioeconomic factors and housing age relate to housing prices in the Boston area.

Data Loading

The Boston Housing dataset contains information about housing values in 506 suburbs of Boston, along with 13 variables that describe different characteristics of these suburbs. Here's a summary of what you've shared:

The dataset has:

- 506 observations (rows)
- 13 variables (columns)
- No missing values

The variables include:

- `crim`: Per capita crime rate by town
- `zn`: Proportion of residential land zoned for lots over 25,000 sq.ft
- `indus`: Proportion of non-retail business acres per town
- `chas`: Charles River dummy variable (1 if tract bounds river; 0 otherwise)
- `nox`: Nitrogen oxides concentration (parts per 10 million)
- `rm`: Average number of rooms per dwelling

- `age`: Proportion of owner-occupied units built prior to 1940
- `dis`: Weighted mean of distances to five Boston employment centers
- `rad`: Index of accessibility to radial highways
- `tax`: Full-value property-tax rate per $10,000
- `ptratio`: Pupil-teacher ratio by town
- `lstat`: Lower status of the population (percent)
- `medv`: Median value of owner-occupied homes in $1000s

```
data(Boston)
glimpse(Boston)

## Rows: 506
## Columns: 13
## $ crim    <dbl> 0.00632, 0.02731, 0.02729, 0.03237, 0.06905, 0.02985,
0.08829,…
## $ zn      <dbl> 18.0, 0.0, 0.0, 0.0, 0.0, 0.0, 12.5, 12.5, 12.5, 12.5,
12.5, 1…
## $ indus   <dbl> 2.31, 7.07, 7.07, 2.18, 2.18, 2.18, 7.87, 7.87, 7.87,
7.87, 7.…
## $ chas    <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0,…
## $ nox     <dbl> 0.538, 0.469, 0.469, 0.458, 0.458, 0.458, 0.524, 0.524,
0.524,…
## $ rm      <dbl> 6.575, 6.421, 7.185, 6.998, 7.147, 6.430, 6.012, 6.172,
5.631,…
## $ age     <dbl> 65.2, 78.9, 61.1, 45.8, 54.2, 58.7, 66.6, 96.1, 100.0,
85.9, 9…
## $ dis     <dbl> 4.0900, 4.9671, 4.9671, 6.0622, 6.0622, 6.0622, 5.5605,
5.9505…
## $ rad     <int> 1, 2, 2, 3, 3, 3, 5, 5, 5, 5, 5, 5, 5, 4, 4, 4, 4, 4, 4,
4, 4,…
## $ tax     <dbl> 296, 242, 242, 222, 222, 222, 311, 311, 311, 311, 311,
311, 31…
## $ ptratio <dbl> 15.3, 17.8, 17.8, 18.7, 18.7, 18.7, 15.2, 15.2, 15.2,
15.2, 15…
## $ lstat   <dbl> 4.98, 9.14, 4.03, 2.94, 5.33, 5.21, 12.43, 19.15, 29.93,
17.10…
## $ medv    <dbl> 24.0, 21.6, 34.7, 33.4, 36.2, 28.7, 22.9, 27.1, 16.5,
18.9, 15…

summary(Boston)

##       crim                  zn              indus            chas
##  Min.   : 0.00632   Min.   :  0.00   Min.   : 0.46   Min.   :0.00000
##  1st Qu.: 0.08205   1st Qu.:  0.00   1st Qu.: 5.19   1st Qu.:0.00000
##  Median : 0.25651   Median :  0.00   Median : 9.69   Median :0.00000
##  Mean   : 3.61352   Mean   : 11.36   Mean   :11.14   Mean   :0.06917
##  3rd Qu.: 3.67708   3rd Qu.: 12.50   3rd Qu.:18.10   3rd Qu.:0.00000
##  Max.   :88.97620   Max.   :100.00   Max.   :27.74   Max.   :1.00000
##       nox               rm             age              dis
##  Min.   :0.3850   Min.   :3.561   Min.   :  2.90   Min.   : 1.130
##  1st Qu.:0.4490   1st Qu.:5.886   1st Qu.: 45.02   1st Qu.: 2.100
##  Median :0.5380   Median :6.208   Median : 77.50   Median : 3.207
##  Mean   :0.5547   Mean   :6.285   Mean   : 68.57   Mean   : 3.795
##  3rd Qu.:0.6240   3rd Qu.:6.623   3rd Qu.: 94.08   3rd Qu.: 5.188
##  Max.   :0.8710   Max.   :8.780   Max.   :100.00   Max.   :12.127
##       rad              tax             ptratio          lstat
##  Min.   : 1.000   Min.   :187.0   Min.   :12.60   Min.   : 1.73
##  1st Qu.: 4.000   1st Qu.:279.0   1st Qu.:17.40   1st Qu.: 6.95
##  Median : 5.000   Median :330.0   Median :19.05   Median :11.36
##  Mean   : 9.549   Mean   :408.2   Mean   :18.46   Mean   :12.65
##  3rd Qu.:24.000   3rd Qu.:666.0   3rd Qu.:20.20   3rd Qu.:16.95
##  Max.   :24.000   Max.   :711.0   Max.   :22.00   Max.   :37.97
##       medv
##  Min.   : 5.00
##  1st Qu.:17.02
##  Median :21.20
##  Mean   :22.53
##  3rd Qu.:25.00
##  Max.   :50.00
```

```r
missing_values = Boston %>%
  summarise(across(everything(), ~ sum(is.na(.))))
print(missing_values)
```

```
##   crim zn indus chas nox rm age dis rad tax ptratio lstat medv
## 1    0  0     0    0   0  0   0   0   0   0       0     0    0
```

Train-Test Split

Train-test splitting is a crucial technique for validating our regression models. When we split our Boston Housing dataset into training and testing sets (typically 70-80% for training, 20-30% for testing), we gain several important benefits for our analysis. First, this approach prevents overfitting by evaluating our model on unseen data, ensuring it generalizes beyond the specific patterns in our training data. The test set provides an honest assessment, giving us an unbiased estimate of how our model will perform on new, unseen Boston housing data. This enables objective model comparison between our simple regression and our multiple regression using the same test data.

```r
set.seed(123) # for reproducibility
Boston_split = Boston %>%
  mutate(id = row_number()) %>%
  sample_frac(0.75)

Boston = Boston %>% mutate(id = row_number())
```

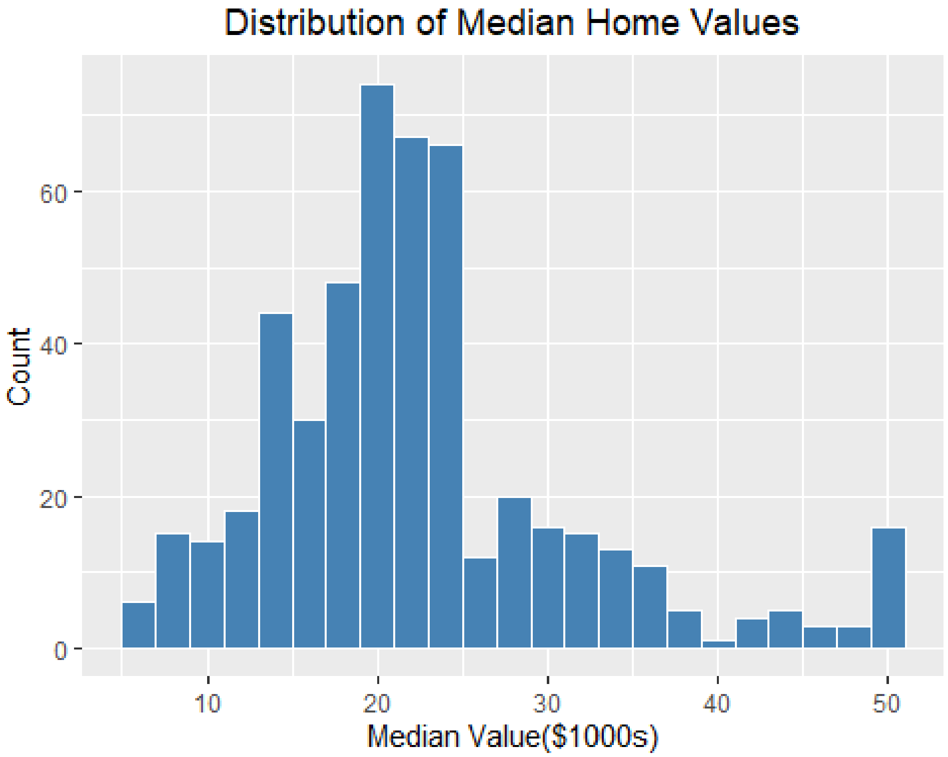
```
train_data = Boston_split
test_data = anti_join(Boston, Boston_split, by ="id") #Remaining 25%
```

Exploratory Data Analysis

In our exploratory data analysis of the Boston Housing dataset, we created two fundamental visualizations to understand the data distribution and relationships.

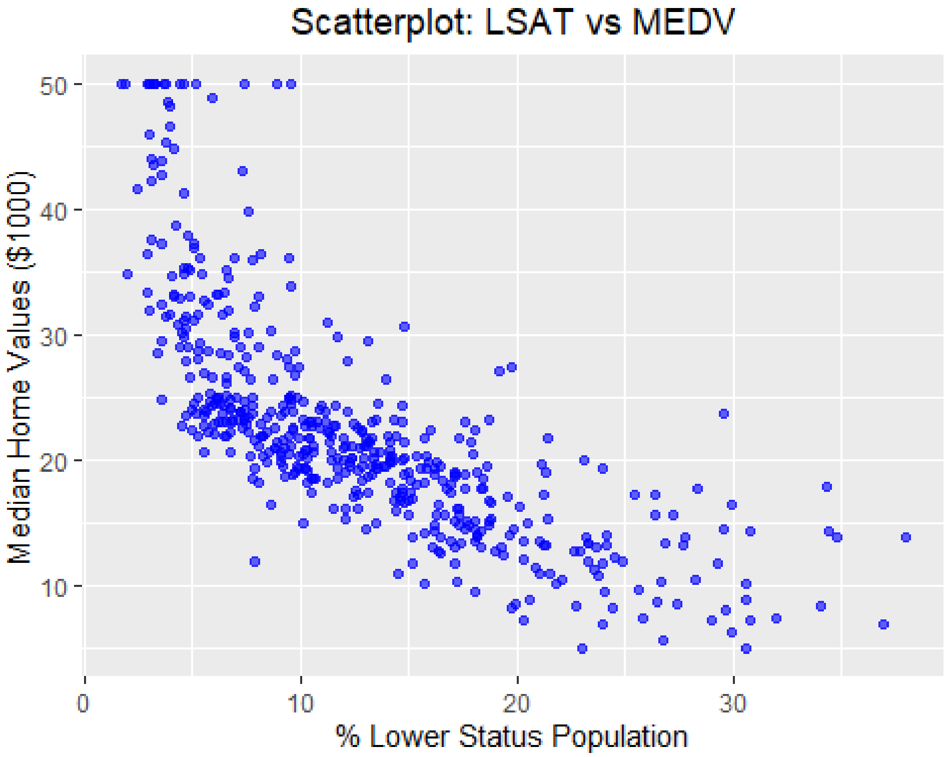
The histogram for median home values (medv) used a binwidth of 2 and was colored in steelblue with white borders for clarity. This visualization revealed the distribution pattern of housing prices across Boston suburbs. The histogram showed that most homes were concentrated in the middle price ranges, but with some notable right-skewness, indicating fewer high-priced outlier neighborhoods. This distribution insight was crucial for understanding our target variable and helped us assess whether transformations might be needed to address non-normality in our regression modeling.

```
ggplot(Boston, aes(x = medv)) +
  geom_histogram(fill = "steelblue", binwidth = 2, color="white") +
  labs(title = "Distribution of Median Home Values",
       x = "Median Value($1000s)",
       y = "Count") +
  theme(plot.title = element_text(hjust = 0.5))
```



Our scatterplot of lower status population percentage (lstat) versus median home values (medv) displayed individual data points in blue with slight transparency (alpha = 0.6) to better visualize point density. This plot revealed a strong negative relationship between these variables - as the percentage of lower status population increases, median home values tend to decrease. Importantly, the scatterplot showed that this relationship is not perfectly linear but appears to have a curved pattern. This visualizations helps us understand the relationship of our target variable with our primary predictor. It provides visual evidence that guided our subsequent modeling decisions.

```r
ggplot(Boston, aes(x=lstat,y=medv)) +
  geom_point(alpha = 0.6 , color = "blue") +
  labs(title = "Scatterplot: LSAT vs MEDV",
       x = "% Lower Status Population",
       y = "Median Home Values ($1000)") +
  theme(plot.title = element_text(hjust = 0.5))
```



Model Implementation & Explanation

For our analysis of the Boston Housing dataset, we implemented linear regression models, both simple and multiple variants. These models are used to predict median home values (medv) based on neighborhood characteristics.

Perform Simple Linear Regession

The primary models include a simple linear regression using only lower status population percentage (lstat) as a predictor.

Linear regression is particularly well-suited for this dataset because there are clear linear relationship between the predictor (lower status population percentage (lstat)) and the target variable, as revealed in our exploratory scatterplots.

```r
lm.fit = lm(medv ~ lstat, data = train_data)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = medv ~ lstat, data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.068  -3.891  -1.275   1.706  24.613
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 34.51650    0.62851   54.92   <2e-16 ***
## lstat       -0.95795    0.04357  -21.99   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.131 on 378 degrees of freedom
## Multiple R-squared:  0.5612, Adjusted R-squared:  0.5601
## F-statistic: 483.5 on 1 and 378 DF,  p-value: < 2.2e-16
```

Apply Model to Test Data

When applying our linear regression model to the test data, we calculated both the training MSE and test MSE to evaluate its performance.

The results from applying our linear regression model to both training and test data reveal that our Training MSE is 37.39 and our Test MSE is 41.86. These values represent the average squared error of our predictions in units of $1000s squared.

The Training MSE of 37.39 indicates that, on average, our model's predictions on the training data are off by approximately $6,115 (√37.39 × $1000). This gives us a baseline for how well our model fits the data it was built upon.

More importantly, the Test MSE of 41.86 shows that when applied to new, unseen data, our model's predictions deviate from actual housing values by about $6,471 (√41.86 × $1000) on average. This represents the true predictive accuracy we can expect when the model encounters new Boston housing data.

The difference between training and test MSE (41.86 - 37.39 = 4.47) is relatively small, suggesting that our model isn't severely overfitting the training data. This is a positive sign that indicates our model generalizes reasonably well to new data points.

However, the fact that the Test MSE is higher than the Training MSE is expected and normal. It confirms the fundamental principle that predictions are typically less accurate on unseen data than on the data used to build the model.

```r
train_mse = mean((train_data$medv - predict(lm.fit, train_data))^2)
test_mse = mean((test_data$medv - predict(lm.fit, test_data))^2)
```

```r
print(paste("Training MSE:", round(train_mse,2)))

## [1] "Training MSE: 37.39"

print(paste("Test MSE: ", round(test_mse,2)))

## [1] "Test MSE:  41.86"
```

Perform Multiple Linear Regression on Training Data

In our analysis, we extended our modeling approach by implementing a multiple linear regression that incorporates both lower status population percentage (lstat) and housing age (age) as predictors of median home values (medv). This allows us to examine how housing age contributes to price prediction beyond the socioeconomic factor.

```r
lm.multiple.fit = lm(medv ~ lstat + age , data = train_data)
summary(lm.multiple.fit)

## 
## Call:
## lm(formula = medv ~ lstat + age, data = train_data)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -15.925  -3.725  -1.214   1.788  23.201 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 33.12191    0.81744  40.519  < 2e-16 ***
## lstat       -1.04394    0.05414 -19.284  < 2e-16 ***
## age          0.03625    0.01374   2.639  0.00867 ** 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 6.083 on 377 degrees of freedom
## Multiple R-squared:  0.5692, Adjusted R-squared:  0.5669 
## F-statistic:   249 on 2 and 377 DF,  p-value: < 2.2e-16
```

Apply the Model to Test Data

```r
train_mse = mean((train_data$medv - predict(lm.multiple.fit, train_data))^2)
test_mse = mean((test_data$medv - predict(lm.multiple.fit, test_data))^2)

print(paste("Training MSE:", round(train_mse,2)))

## [1] "Training MSE: 36.72"

print(paste("Test MSE: ", round(test_mse,2)))

## [1] "Test MSE:  41.49"
```

Multiple Linear Regression Results & Interpretation

The coefficient for lstat is -1.04394, which means that for each one percentage point increase in lower status population, the median home value decreases by approximately $1,044, holding housing age constant. This strong negative relationship is highly statistically significant (p < 2e-16), confirming that socioeconomic factors strongly influence housing prices.

Interestingly, the coefficient for age is positive (0.03625), indicating that each additional percentage point of pre-1940 housing is associated with a slight increase of about $36 in median home value when controlling for socioeconomic status. This positive relationship is statistically significant (p = 0.00867), though the effect size is much smaller than that of lstat. This contradicts what might be expected - older housing stock typically correlates with lower values - suggesting that in Boston, older homes may have historical value or be located in desirable established neighborhoods when socioeconomic factors are controlled for.

The model explains approximately 57% of the variance in housing prices (R-squared = 0.5692), which represents substantial explanatory power. The F-statistic is highly significant (p < 2.2e-16), confirming that our model as a whole provides meaningful predictions.

Looking at performance metrics, the Training MSE decreased slightly from 37.39 in our simple model to 36.72 in this multiple regression model. Similarly, the Test MSE improved from 41.86 to 41.49. This modest improvement (about 0.9% reduction in test error) suggests that while age does add some predictive power, its contribution is relatively small compared to the socioeconomic factor measured by lstat.

Overall, while adding the age variable does improve our model, the improvement is modest. The socioeconomic status of the neighborhood remains the dominant factor in predicting Boston housing values.

NHANES Data Analysis

Objective

The goal of this analysis is to develop a multiple regression model to predict Body Mass Index (BMI) using data from the National Health and Nutrition Examination Survey (NHANES). The model incorporates three predictor variables: age, smoking habits, and physical activity for individuals between the age of 18 and 70. This approach aims to provide insights into how these factors collectively influence BMI, enhancing our understanding of their relationships within the context of public health.

Data Loading

```r
library(NHANES)

## Warning: package 'NHANES' was built under R version 4.3.3
```

```r
data(NHANES)
str(NHANES)
```

```
## tibble [10,000 × 76] (S3: tbl_df/tbl/data.frame)
##  $ ID              : int [1:10000] 51624 51624 51624 51625 51630 51638
51646 51647 51647 51647 ...
##  $ SurveyYr        : Factor w/ 2 levels "2009_10","2011_12": 1 1 1 1 1 1 1
1 1 1 ...
##  $ Gender          : Factor w/ 2 levels "female","male": 2 2 2 2 1 2 2 1 1
1 ...
##  $ Age             : int [1:10000] 34 34 34 4 49 9 8 45 45 45 ...
##  $ AgeDecade       : Factor w/ 8 levels " 0-9"," 10-19",..: 4 4 4 1 5 1 1
5 5 5 ...
##  $ AgeMonths       : int [1:10000] 409 409 409 49 596 115 101 541 541 541
...
##  $ Race1           : Factor w/ 5 levels "Black","Hispanic",..: 4 4 4 5 4 4
4 4 4 4 ...
##  $ Race3           : Factor w/ 6 levels "Asian","Black",..: NA NA NA NA NA
NA NA NA NA NA ...
##  $ Education       : Factor w/ 5 levels "8th Grade","9 - 11th Grade",..: 3
3 3 NA 4 NA NA 5 5 5 ...
##  $ MaritalStatus   : Factor w/ 6 levels "Divorced","LivePartner",..: 3 3 3
NA 2 NA NA 3 3 3 ...
##  $ HHIncome        : Factor w/ 12 levels " 0-4999"," 5000-9999",..: 6 6 6
5 7 11 9 11 11 11 ...
##  $ HHIncomeMid     : int [1:10000] 30000 30000 30000 22500 40000 87500
60000 87500 87500 87500 ...
##  $ Poverty         : num [1:10000] 1.36 1.36 1.36 1.07 1.91 1.84 2.33 5 5
5 ...
##  $ HomeRooms       : int [1:10000] 6 6 6 9 5 6 7 6 6 6 ...
##  $ HomeOwn         : Factor w/ 3 levels "Own","Rent","Other": 1 1 1 1 2 2
1 1 1 1 ...
##  $ Work            : Factor w/ 3 levels "Looking","NotWorking",..: 2 2 2
NA 2 NA NA 3 3 3 ...
##  $ Weight          : num [1:10000] 87.4 87.4 87.4 17 86.7 29.8 35.2 75.7
75.7 75.7 ...
##  $ Length          : num [1:10000] NA NA NA NA NA NA NA NA NA NA ...
##  $ HeadCirc        : num [1:10000] NA NA NA NA NA NA NA NA NA NA ...
##  $ Height          : num [1:10000] 165 165 165 105 168 ...
##  $ BMI             : num [1:10000] 32.2 32.2 32.2 15.3 30.6 ...
##  $ BMICatUnder20yrs: Factor w/ 4 levels "UnderWeight",..: NA NA NA NA NA
NA NA NA NA NA ...
##  $ BMI_WHO         : Factor w/ 4 levels "12.0_18.5","18.5_to_24.9",..: 4 4
4 1 4 1 2 3 3 3 ...
##  $ Pulse           : int [1:10000] 70 70 70 NA 86 82 72 62 62 62 ...
##  $ BPSysAve        : int [1:10000] 113 113 113 NA 112 86 107 118 118 118
...
##  $ BPDiaAve        : int [1:10000] 85 85 85 NA 75 47 37 64 64 64 ...
##  $ BPSys1          : int [1:10000] 114 114 114 NA 118 84 114 106 106 106
...
```

```
##  $ BPDia1         : int [1:10000] 88 88 88 NA 82 50 46 62 62 62 ...
##  $ BPSys2         : int [1:10000] 114 114 114 NA 108 84 108 118 118 118
...
##  $ BPDia2         : int [1:10000] 88 88 88 NA 74 50 36 68 68 68 ...
##  $ BPSys3         : int [1:10000] 112 112 112 NA 116 88 106 118 118 118
...
##  $ BPDia3         : int [1:10000] 82 82 82 NA 76 44 38 60 60 60 ...
##  $ Testosterone    : num [1:10000] NA NA NA NA NA NA NA NA NA NA ...
##  $ DirectChol      : num [1:10000] 1.29 1.29 1.29 NA 1.16 1.34 1.55 2.12
2.12 2.12 ...
##  $ TotChol         : num [1:10000] 3.49 3.49 3.49 NA 6.7 4.86 4.09 5.82
5.82 5.82 ...
##  $ UrineVol1       : int [1:10000] 352 352 352 NA 77 123 238 106 106 106
...
##  $ UrineFlow1      : num [1:10000] NA NA NA NA 0.094 ...
##  $ UrineVol2       : int [1:10000] NA NA NA NA NA NA NA NA NA NA ...
##  $ UrineFlow2      : num [1:10000] NA NA NA NA NA NA NA NA NA NA ...
##  $ Diabetes        : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1
...
##  $ DiabetesAge     : int [1:10000] NA NA NA NA NA NA NA NA NA NA ...
##  $ HealthGen       : Factor w/ 5 levels "Excellent","Vgood",..: 3 3 3 NA 3
NA NA 2 2 2 ...
##  $ DaysPhysHlthBad : int [1:10000] 0 0 0 NA 0 NA NA 0 0 0 ...
##  $ DaysMentHlthBad : int [1:10000] 15 15 15 NA 10 NA NA 3 3 3 ...
##  $ LittleInterest  : Factor w/ 3 levels "None","Several",..: 3 3 3 NA 2 NA
NA 1 1 1 ...
##  $ Depressed       : Factor w/ 3 levels "None","Several",..: 2 2 2 NA 2 NA
NA 1 1 1 ...
##  $ nPregnancies    : int [1:10000] NA NA NA NA 2 NA NA 1 1 1 ...
##  $ nBabies         : int [1:10000] NA NA NA NA 2 NA NA NA NA NA ...
##  $ Age1stBaby      : int [1:10000] NA NA NA NA 27 NA NA NA NA NA ...
##  $ SleepHrsNight   : int [1:10000] 4 4 4 NA 8 NA NA 8 8 8 ...
##  $ SleepTrouble    : Factor w/ 2 levels "No","Yes": 2 2 2 NA 2 NA NA 1 1 1
...
##  $ PhysActive      : Factor w/ 2 levels "No","Yes": 1 1 1 NA 1 NA NA 2 2 2
...
##  $ PhysActiveDays  : int [1:10000] NA NA NA NA NA NA NA 5 5 5 ...
##  $ TVHrsDay        : Factor w/ 7 levels "0_hrs","0_to_1_hr",..: NA NA NA
NA NA NA NA NA NA NA ...
##  $ CompHrsDay      : Factor w/ 7 levels "0_hrs","0_to_1_hr",..: NA NA NA
NA NA NA NA NA NA NA ...
##  $ TVHrsDayChild   : int [1:10000] NA NA NA 4 NA 5 1 NA NA NA ...
##  $ CompHrsDayChild : int [1:10000] NA NA NA 1 NA 0 6 NA NA NA ...
##  $ Alcohol12PlusYr : Factor w/ 2 levels "No","Yes": 2 2 2 NA 2 NA NA 2 2 2
...
##  $ AlcoholDay      : int [1:10000] NA NA NA NA 2 NA NA 3 3 3 ...
##  $ AlcoholYear     : int [1:10000] 0 0 0 NA 20 NA NA 52 52 52 ...
##  $ SmokeNow        : Factor w/ 2 levels "No","Yes": 1 1 1 NA 2 NA NA NA NA
NA ...
##  $ Smoke100        : Factor w/ 2 levels "No","Yes": 2 2 2 NA 2 NA NA 1 1 1
```

```
...
##  $ Smoke100n       : Factor w/ 2 levels "Non-Smoker","Smoker": 2 2 2 NA 2
NA NA 1 1 1 ...
##  $ SmokeAge        : int [1:10000] 18 18 18 NA 38 NA NA NA NA NA ...
##  $ Marijuana       : Factor w/ 2 levels "No","Yes": 2 2 2 NA 2 NA NA 2 2 2
...
##  $ AgeFirstMarij   : int [1:10000] 17 17 17 NA 18 NA NA 13 13 13 ...
##  $ RegularMarij    : Factor w/ 2 levels "No","Yes": 1 1 1 NA 1 NA NA 1 1 1
...
##  $ AgeRegMarij     : int [1:10000] NA NA NA NA NA NA NA NA NA NA ...
##  $ HardDrugs       : Factor w/ 2 levels "No","Yes": 2 2 2 NA 2 NA NA 1 1 1
...
##  $ SexEver         : Factor w/ 2 levels "No","Yes": 2 2 2 NA 2 NA NA 2 2 2
...
##  $ SexAge          : int [1:10000] 16 16 16 NA 12 NA NA 13 13 13 ...
##  $ SexNumPartnLife : int [1:10000] 8 8 8 NA 10 NA NA 20 20 20 ...
##  $ SexNumPartYear  : int [1:10000] 1 1 1 NA 1 NA NA 0 0 0 ...
##  $ SameSex         : Factor w/ 2 levels "No","Yes": 1 1 1 NA 2 NA NA 2 2 2
...
##  $ SexOrientation  : Factor w/ 3 levels "Bisexual","Heterosexual",..: 2 2
2 NA 2 NA NA 1 1 1 ...
##  $ PregnantNow     : Factor w/ 3 levels "Yes","No","Unknown": NA NA NA NA
NA NA NA NA NA NA ...
```

Data Understanding & Preperation

We create a customized dataset named SMOKERS for our analysis, extracting key variables; BMI (Body Mass Index), Age, SmokeNow (current smoking status), and PhysActive (physical activity level), while limiting the data to individuals between 18 and 70 years old. Upon exploring the SMOKERS dataset, we discover that 58% of the SmokeNow values are missing, while only 0.7% of the BMI data is absent. To address the substantial missing data in SmokeNow, we opt for mode imputation to fill in these gaps, recognizing the importance of correcting for such a large proportion of missing values. For the BMI variable, given the minimal missing data, we choose to remove those records entirely.

```r
# Load the dplyr package (or the entire tidyverse)
#library(dplyr)

# Then run your code
SMOKERS = NHANES %>%
  select(BMI, Age, SmokeNow, PhysActive) %>%
  filter(Age >= 18 & Age <= 70)

str(SMOKERS)

## tibble [6,663 × 4] (S3: tbl_df/tbl/data.frame)
##  $ BMI       : num [1:6663] 32.2 32.2 32.2 30.6 27.2 ...
##  $ Age       : int [1:6663] 34 34 34 49 45 45 45 66 58 54 ...
##  $ SmokeNow  : Factor w/ 2 levels "No","Yes": 1 1 1 2 NA NA NA 1 NA NA ...
##  $ PhysActive: Factor w/ 2 levels "No","Yes": 1 1 1 1 2 2 2 2 2 2 ...
```

```
unique(SMOKERS[, c("PhysActive", "SmokeNow")])

## # A tibble: 6 × 2
##   PhysActive SmokeNow
##   <fct>      <fct>
## 1 No         No
## 2 No         Yes
## 3 Yes        <NA>
## 4 Yes        No
## 5 Yes        Yes
## 6 No         <NA>

# Count missing values for each column
colSums(is.na(SMOKERS))

##        BMI        Age   SmokeNow PhysActive
##         47          0       3868          0

# percentage of missing values
colMeans(is.na(SMOKERS))

##        BMI        Age   SmokeNow PhysActive
## 0.00705388 0.00000000 0.58051929 0.00000000
```

“mode imputation” - replacing missing values with the most common value in the dataset.

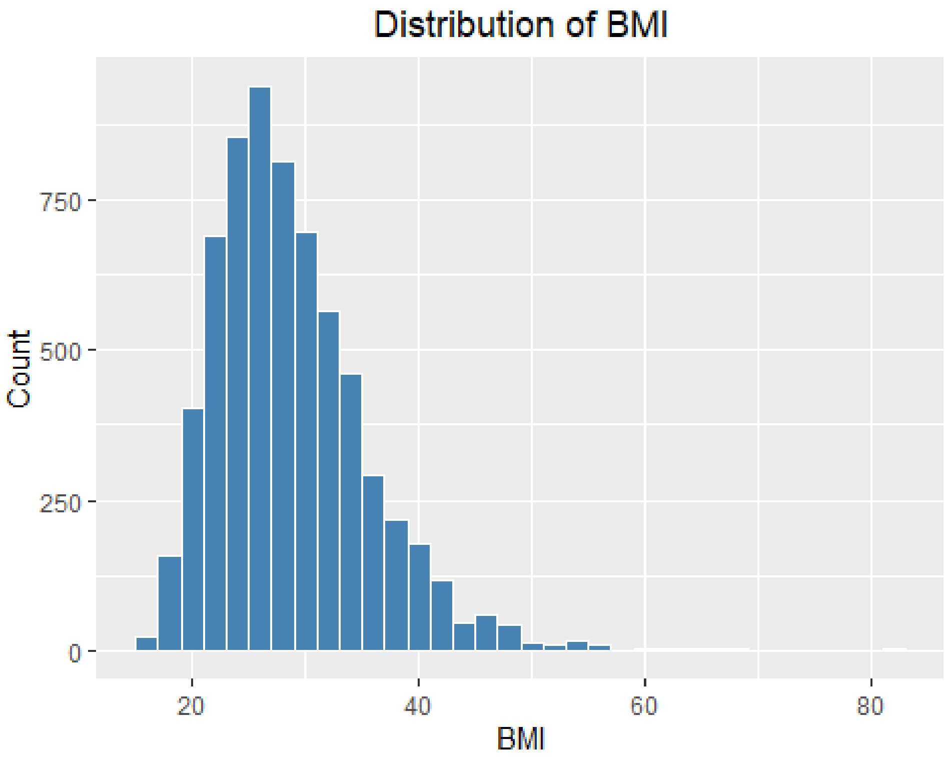
```
# Or if it's a categorical variable, replace with the most frequent value:
SMOKERS$SmokeNow[is.na(SMOKERS$SmokeNow)] <-
names(which.max(table(SMOKERS$SmokeNow, useNA = "no")))

colSums(is.na(SMOKERS))

##        BMI        Age   SmokeNow PhysActive
##         47          0          0          0

# drop the missing values in BMI
library(dplyr)
SMOKERS <- SMOKERS %>% filter(!is.na(BMI))

colSums(is.na(SMOKERS))

##        BMI        Age   SmokeNow PhysActive
##          0          0          0          0
```

Exploratory Data Analysis

In our exploratory data analysis of the SMOKERS dataset, we created two fundamental visualizations to understand the data distribution and relationships.

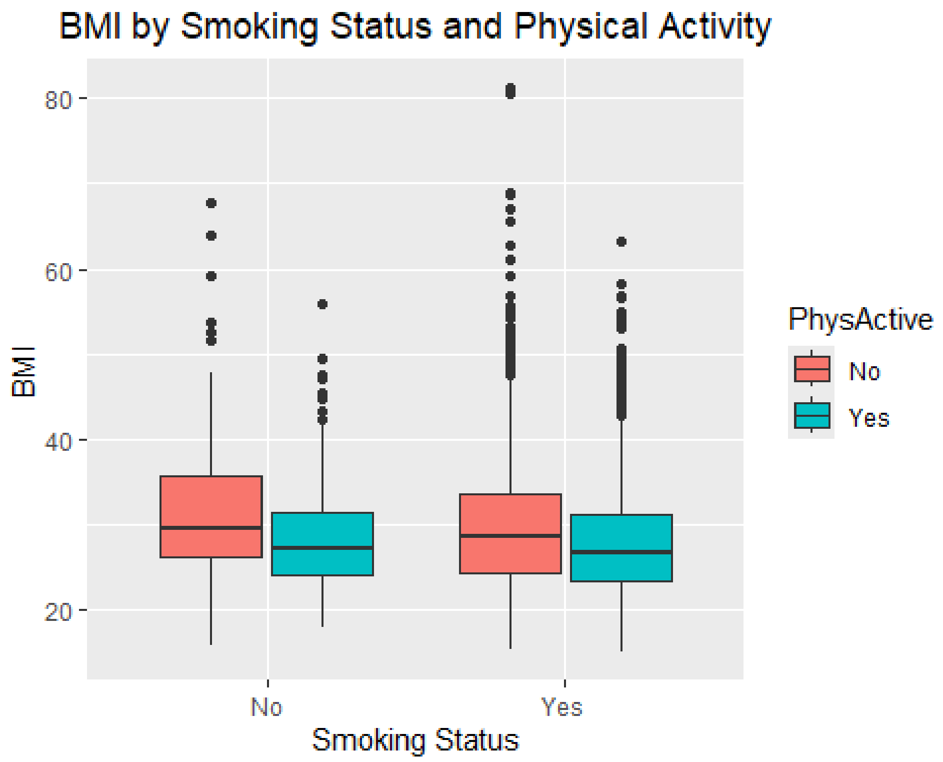
The histogram distribution of BMI shows a right-skewed pattern, with the majority of individuals clustered between 20 and 40 BMI, peaking around 25–30, which suggests a concentration of participants in the overweight category. The tail extends toward higher BMI values, with fewer individuals exceeding 40, indicating a smaller proportion of obese individuals. This skewed distribution may reflect underlying factors such as smoking status or physical activity levels, which could be further explored with the dataset's additional variables.

```r
ggplot(SMOKERS, aes(x = BMI)) +
  geom_histogram(fill = "steelblue", binwidth = 2, color="white") +
  labs(title = "Distribution of BMI",
       x = "BMI",
       y = "Count") +
  theme(plot.title = element_text(hjust = 0.5))
```



The boxplot reveals significant relationships between BMI, smoking status, and physical activity.

```r
ggplot(SMOKERS, aes(x = SmokeNow, y = BMI, fill = PhysActive)) +
  geom_boxplot() +
  labs(title = "BMI by Smoking Status and Physical Activity",
       x = "Smoking Status",
       y = "BMI") +
  theme(plot.title = element_text(hjust = 0.5))
```



Physically active individuals (represented by teal boxes) consistently demonstrate lower BMI values than their physically inactive counterparts (shown in red boxes) across both smoking categories. The median BMI for physically active participants is approximately 25-27, while physically inactive individuals show median BMI values around 30-32. Notably, non-smokers who are physically inactive exhibit the highest median BMI, with their interquartile range showing greater variability than other groups. Several outliers appear in the upper BMI range (60-80), particularly among physically inactive individuals, suggesting extreme cases that deviate from the general pattern. This visualization supports the potential effectiveness of a multiple linear regression model using smoking status (SmokeNow) and physical activity (PhysActive) as predictors of BMI. The clear separation between physically active and inactive groups indicates that physical activity likely has a stronger association with BMI than smoking status (as physical activity increases, BMI tends to decrease) , though both variables appear to contribute to BMI variations.

Train-Test Split

The provided code implements a crucial data preprocessing technique called train-test splitting for the SMOKERS dataset. 75% of the data is randomly chosen for training the model. The remaining 25% is kept aside for testing. This way, the model learns from the training data and is then evaluated on new, unseen data. This separation is important because it helps ensure that the model doesn't just memorize the training data (overfitting), but can actually perform well when making predictions on different data. It also provides a realistic assessment of how well the model will work in real-world situations, confirming that the relationships between factors like age, smoking status, physical activity, and BMI remain consistent.

```r
# Set seed for reproducibility
set.seed(123)

# Create a data split (75% train, 25% test)
SMOKERS_split = SMOKERS %>%
  mutate(id = row_number()) %>%
  sample_frac(0.75)

# Add id to original dataset
SMOKERS = SMOKERS %>% mutate(id = row_number())

# Create train and test datasets
train_data2 = SMOKERS_split
test_data2 = anti_join(SMOKERS, SMOKERS_split, by = "id") # Remaining 25%
```

Model Implementation & Explanation

For our analysis of the SMOKERS dataset, we implemented multiple linear regression model. This model is used to predict the Body Mass Index (BMI) using three predictor variables: age, smoking habits, and physical activity.

Perform Multiple Linear Regession

Multiple linear regression is a statistical method used to model the relationship between one dependent variable (here, BMI) and multiple independent variables (Age, SmokeNow, PhysActive). The goal is to find a linear equation that best predicts BMI based on these predictors. This analysis helps us understand how Age, smoking habits (SmokeNow), and physical activity (PhysActive) collectively influence BMI.

```r
# Fit linear regression model on training data
model = lm(BMI ~ Age + SmokeNow + PhysActive, data = train_data2)

# View model summary
summary(model)

## 
## Call:
## lm(formula = BMI ~ Age + SmokeNow + PhysActive, data = train_data2)
## 
## Residuals:
##     Min       1Q  Median       3Q     Max
## -13.773  -4.615  -1.243   3.564  51.769
## 
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   27.676394   0.415766  66.567  < 2e-16 ***
## Age            0.051224   0.006779   7.556 4.92e-14 ***
## SmokeNowYes   -0.125525   0.242205  -0.518    0.604
## PhysActiveYes -1.769843   0.194391  -9.105  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.746 on 4958 degrees of freedom
## Multiple R-squared:  0.03184,    Adjusted R-squared:  0.03125
## F-statistic: 54.35 on 3 and 4958 DF,  p-value: < 2.2e-16

# Calculate MSE for training data
train_mse2 = mean((train_data2$BMI - predict(model, train_data2))^2)

# Calculate MSE for test data
test_mse2 = mean((test_data2$BMI - predict(model, test_data2))^2)

# Print results
print(paste("Training MSE:", round(train_mse2, 2)))

## [1] "Training MSE: 45.47"

print(paste("Test MSE:", round(test_mse2, 2)))

## [1] "Test MSE: 43.31"

# You may also want to calculate R-squared for both sets
train_r2 = summary(model)$r.squared
# For test data, calculate R-squared manually
y_test = test_data2$BMI
y_pred = predict(model, test_data2)
test_r2 = 1 - (sum((y_test - y_pred)^2) / sum((y_test - mean(y_test))^2))

print(paste("Training R-squared:", round(train_r2, 4)))

## [1] "Training R-squared: 0.0318"

print(paste("Test R-squared:", round(test_r2, 4)))

## [1] "Test R-squared: 0.0292"
```

Results & Discussion

- **Model Performance**:

  - Training MSE: 45.47
  - Test MSE: 43.31
  - Training R-squared: 0.0318
  - Test R-squared: ~0.034

  The test error is slightly lower than the training error, indicating no major overfitting. However, the low R-squared values show that only around 3% of the variation in BMI is explained.

- **Key Model Results**:

  - **Intercept**: 27.68
  - **Age**: Coefficient = 0.051 (p < 0.001) — BMI increases slightly with age.
  - **Physical Activity**: Coefficient for “Yes” = -1.77 (p < 0.001) — Being physically active is associated with a lower BMI.
  - **Smoking Status**: Coefficient for “Yes” = -0.126 (p = 0.604) — No significant effect on BMI.

- **Interpretation**:
  - Physical activity appears to be a meaningful predictor of lower BMI.
  - Age has a modest positive effect on BMI.
  - Smoking status shows no clear link to BMI in this model, which may be due to high missing data (58% imputed).
  - A **Residual Standard Error** of 6.746 means the model’s BMI predictions can be off by a fair margin.

- **Healthcare Context**:
  - The low R-squared (3.18%) suggests many other factors (e.g., diet, genetics) are important for explaining BMI.
  - These findings provide a starting point, but more comprehensive data and variables are needed for stronger predictive power in public health settings.