Linear Regression

CCE20005: Introduction to Data Science

Presented by Hyebong Choi

Contents

- Concept
- Example

What is Regression?

Regression: Predict a numerical outcome ("dependent variable") from a set of inputs ("independent variables").

- ✓ Statistical Sense : Predicting the expected value of the outcome.
- ✓ Casual Sense : Predicting a numerical outcome, rather than a discrete one.

- ✓ How many units will we sell? **(Regression)**
- ✓ Will this customer buy our product (yes/no)? **(Classification)**
- ✓ What price will the customer pay for our product? **(Regression)**

Regression from a Machine Learning Perspective

- ✓ Scientific mindset : Modeling to understand the data generation process
- ✓ Engineering mindset : *Modeling to predict accurately

Machine Learning: Engineering mindset

Linear Regression Hypothesis

Linearity Assumption:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots$$

y is *linearly* related to each x_i

Each x_i contributes additively to *y*

Simplified Example

```
unemployment <-
read.csv('https://github.com/hbchoi/SampleData/raw/master/unemployment.csv')

unemployment

##    male_unemployment female_unemployment
## 1                2.9                 4.0
## 2                6.7                 7.4
## 3                4.9                 5.0
## 4                7.9                 7.2
## 5                9.8                 7.9
## 6                6.9                 6.1
## 7                6.1                 6.0
## 8                6.2                 5.8
## 9                6.0                 5.2
## 10               5.1                 4.2
## 11               4.7                 4.0
## 12               4.4                 4.4
## 13               5.8                 5.2
```

the dataset contains the rates of male and female unemployment in the United States over several years

we assume that **female unemployment rate *y*** is *linearly* related to **male unemployment rate *x***

$$\hat{y} = \alpha x$$

where $\hat{y}$ is estimated outcome y

Simplified Example

```
# set alpha = 1
alpha <- 1
unemployment$est_y <- alpha * unemployment$male_unemployment
unemployment$error <- unemployment$female_unemployment - unemployment$est_y
unemployment
```

```
##    male_unemployment female_unemployment est_y      error
## 1                2.9                 4.0   2.9  1.0999999
## 2                6.7                 7.4   6.7  0.7000003
## 3                4.9                 5.0   4.9  0.0999999
## 4                7.9                 7.2   7.9 -0.7000003
## 5                9.8                 7.9   9.8 -1.9000001
## 6                6.9                 6.1   6.9 -0.8000002
## 7                6.1                 6.0   6.1 -0.0999999
## 8                6.2                 5.8   6.2 -0.3999996
## 9                6.0                 5.2   6.0 -0.8000002
## 10               5.1                 4.2   5.1 -0.9000001
## 11               4.7                 4.0   4.7 -0.6999998
## 12               4.4                 4.4   4.4  0.0000000
## 13               5.8                 5.2   5.8 -0.6000004
```

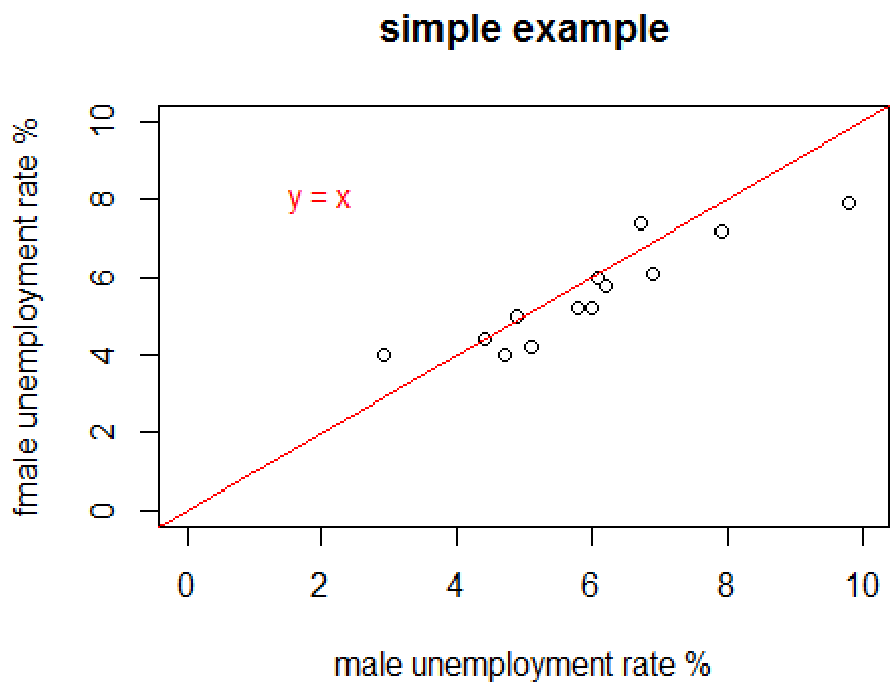
Let us randomly choose any value for $\boldsymbol{\alpha} = 1$ then error will be $\mathrm{y} - \hat{\boldsymbol{y}}$

```
## mean of squared error
mse = mean(unemployment$error ** 2)
mse
```

```
## [1] 0.686154
```

Simplified Example

```
plot(x=unemployment$male_unemployment,
     y=unemployment$female_unemployment,
     main = 'simple example',
     xlab = 'male unemployment rate %',
     ylab = 'fmale unemployment rate %',
     xlim = c(0,10), ylim = c(0,10))
abline(0, alpha, col = 'red')
text(x= 2, y= 8, 'y = x', col = 'red')
```



Try different alpha

```
# set alpha = 0.9
alpha <- 0.9
unemployment$est_y <- alpha * unemployment$male_unemployment
unemployment$error <- unemployment$female_unemployment - unemployment$est_y
unemployment
```

```
##    male_unemployment female_unemployment est_y       error
## 1                2.9                 4.0  2.61  1.38999991
## 2                6.7                 7.4  6.03  1.37000027
## 3                4.9                 5.0  4.41  0.58999991
## 4                7.9                 7.2  7.11  0.08999972
## 5                9.8                 7.9  8.82 -0.92000008
## 6                6.9                 6.1  6.21 -0.11000018
## 7                6.1                 6.0  5.49  0.51000009
## 8                6.2                 5.8  5.58  0.22000036
## 9                6.0                 5.2  5.40 -0.20000019
## 10               5.1                 4.2  4.59 -0.39000011
## 11               4.7                 4.0  4.23 -0.22999983
## 12               4.4                 4.4  3.96  0.44000001
## 13               5.8                 5.2  5.22 -0.02000036
```

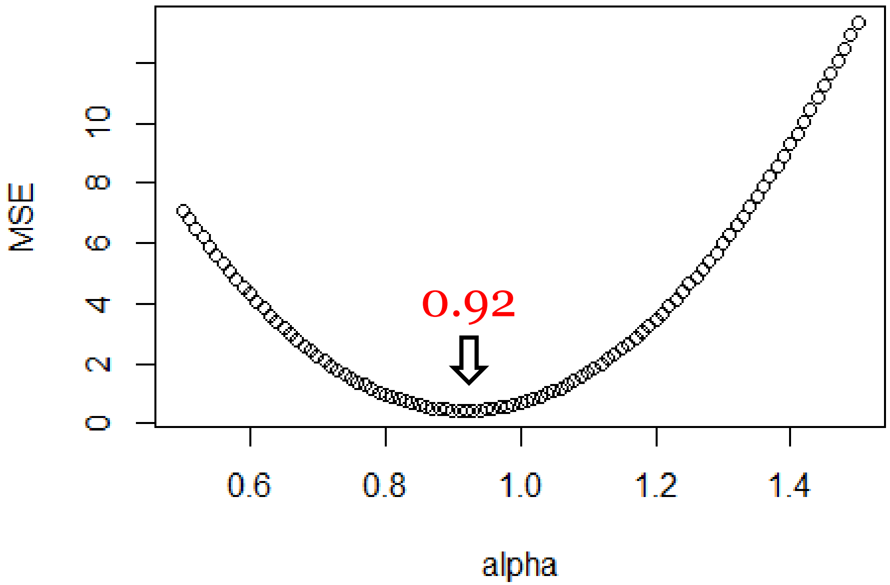
```
## mean of squared error
mse = mean(unemployment$error ** 2)
mse
```

```
## [1] 0.4439385
```

MSE changes over alpha

```
findMSE <- function(alpha){
  mse <- mean((unemployment$female_unemployment -
unemployment$male_unemployment * alpha ) ** 2)
}

alpha_list <- seq(0.5,1.5,0.01)
MSE_list <- sapply(alpha_list, findMSE)

plot(x=alpha_list, y=MSE_list, xlab = 'alpha', ylab = 'MSE')
best_alpha <- alpha_list[which.min(MSE_list)]
best_alpha
```

```
## [1] 0.92
```



Simplified Example

```
plot(x=unemployment$male_unemployment,
     y=unemployment$female_unemployment,
     main = 'simple example',
     xlab = 'male unemployment rate %',
     ylab = 'fmale unemployment rate %',
     xlim = c(0,10), ylim = c(0,10))
abline(0, best_alpha, col = 'red')
text(x= 2, y= 8, 'y = 0.92*x', col = 'red')
```

What’s missing?

bias term (y intercept)

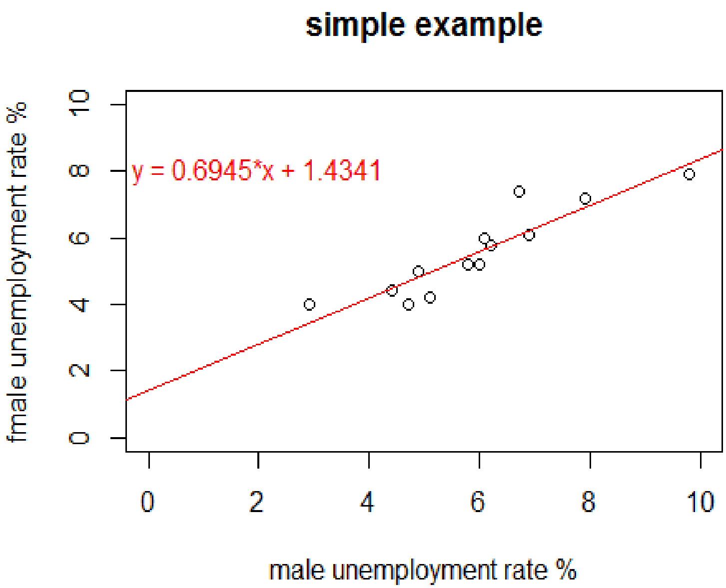
```r
plot(x=unemployment$male_unemployment,
     y=unemployment$female_unemployment,
     main = 'simple example',
     xlab = 'male unemployment rate %',
     ylab = 'fmale unemployment rate %',
     xlim = c(0,10), ylim = c(0,10))
abline(1.4341, 0.6945, col = 'red')
text(x= 2, y= 8, 'y = 0.6945*x + 1.4341', col =
'red')
```

```r
alpha <- 0.6945
beta <- 1.4341
unemployment$est_y <- alpha * unemployment$male_unemployment + beta
unemployment$error <- unemployment$female_unemployment - unemployment$est_y
unemployment
```

```
##    male_unemployment female_unemployment   est_y       error
## 1                2.9                 4.0 3.44815  0.55184993
## 2                6.7                 7.4 6.08725  1.31275023
## 3                4.9                 5.0 4.83715  0.16284993
## 4                7.9                 7.2 6.92065  0.27934974
## 5                9.8                 7.9 8.24020 -0.34020004
## 6                6.9                 6.1 6.22615 -0.12615016
## 7                6.1                 6.0 5.67055  0.32945007
## 8                6.2                 5.8 5.74000  0.06000032
## 9                6.0                 5.2 5.60110 -0.40110019
## 10               5.1                 4.2 4.97605 -0.77605013
## 11               4.7                 4.0 4.69825 -0.69824987
## 12               4.4                 4.4 4.48990 -0.08989997
## 13               5.8                 5.2 5.46220 -0.26220032
```

```r
## mean of squared error
mse = mean(unemployment$error ** 2)
mse
```

```
## [1] 0.2849011
```



Formulas

```
> fmla_1 <- temperature ~ chirps_per_sec
> fmla_2 <- blood_pressure ~ age + weight
```

- ✓ LHS: outcome
- ✓ RHS: inputs
  - ✓ use + for multiple inputs

```
> fmla_1 <- as.formula("temperature ~ chirps_per_sec")
```

```
model <- lm(formula, data = data frame)
```

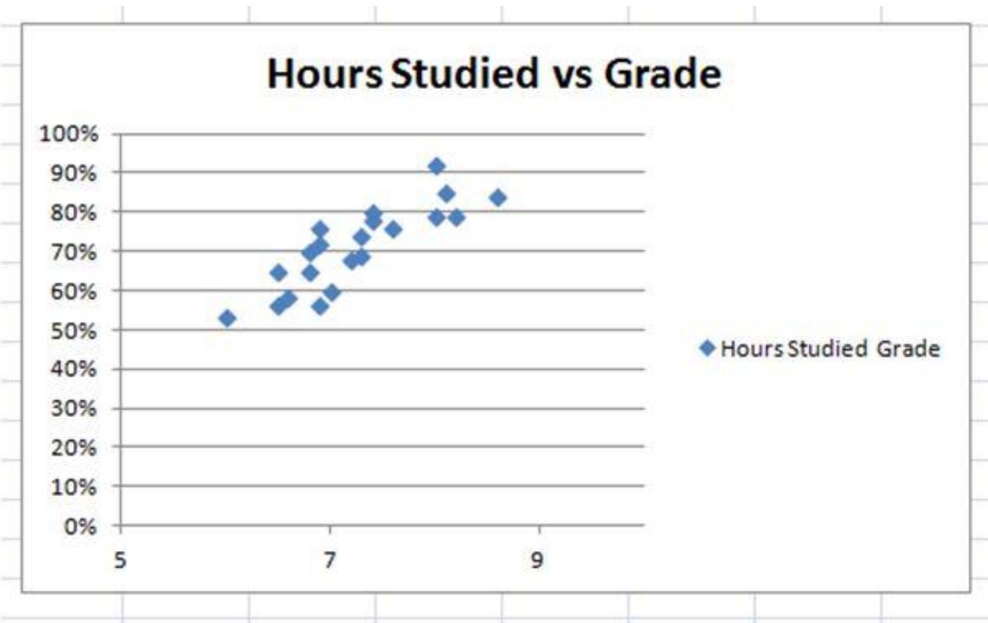
Finding a model that minimizes MSE

```
fmla <- female_unemployment ~ male_unemployment
unemployment_model <- lm(fmla, data = unemployment)
unemployment_model
```

```
##
## Call:
## lm(formula = fmla, data = unemployment)
##
## Coefficients:
##       (Intercept)  male_unemployment
##            1.4341             0.6945
```

Linear Regression

- Basic Idea



Y (Grade)	X (Hours Studied)
70	6
78	7
75	6.5
88	8
…	…

assumption

Dependent Variable

Population Y intercept

Population Slope Coefficient

Independent Variable

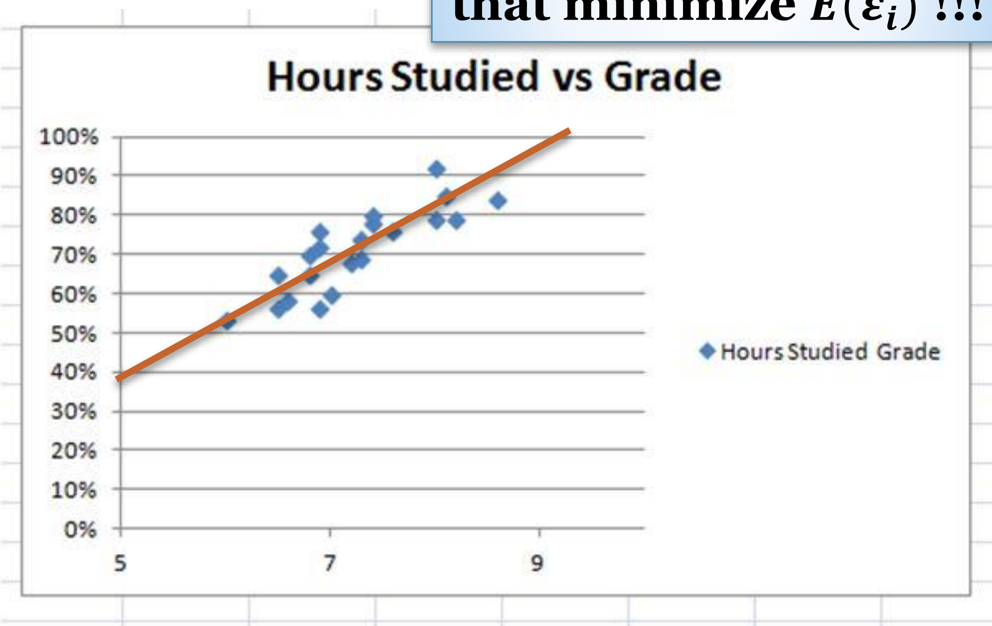
Random Error term

$$Y_i = \underbrace{\beta_0 + \beta_1 X_i}_{\text{Linear component}} + \underbrace{\varepsilon_i}_{\text{Random Error component}}$$

Linear Regression

- Basic Idea

find β_0, β_1
that minimize $E(\varepsilon_i)$!!!



Y (Grade)	X (Hours Studied)
70	6
78	7
75	6.5
88	8
…	…

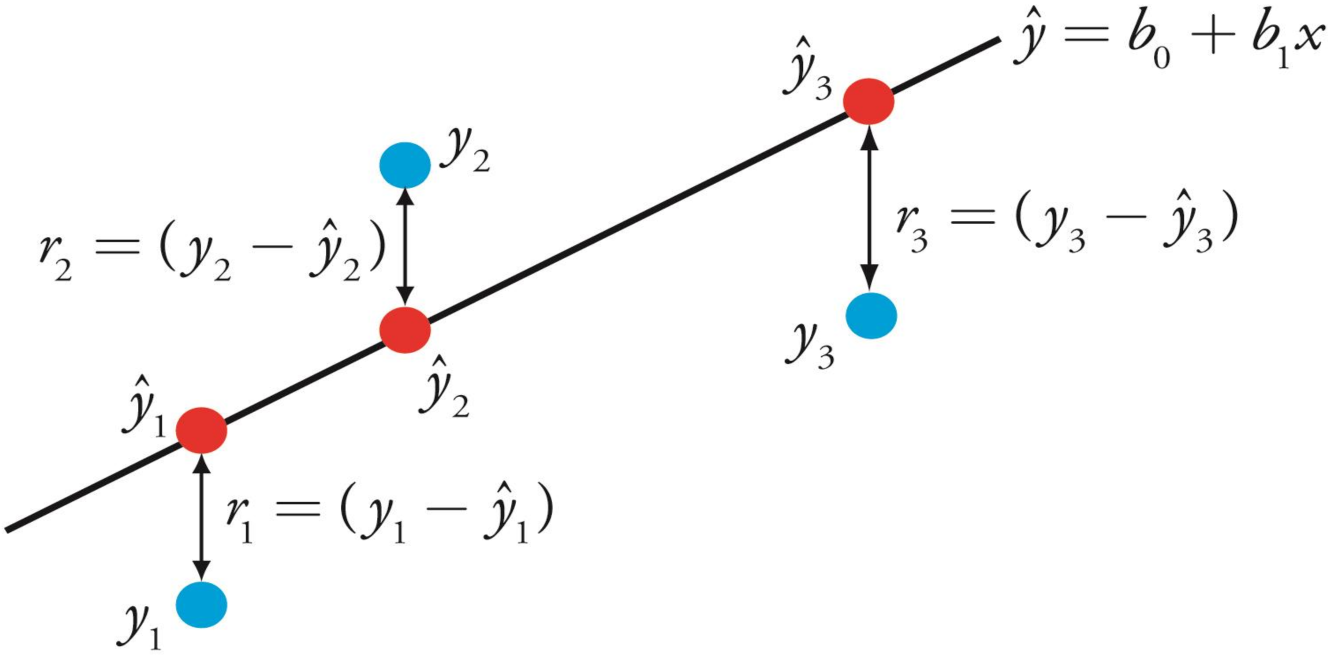
assumption

Dependent Variable · Population Y intercept · Population Slope Coefficient · Independent Variable · Random Error term

$$Y_i = \underbrace{\beta_0 + \beta_1 X_i}_{\text{Linear component}} + \underbrace{\varepsilon_i}_{\text{Random Error component}}$$

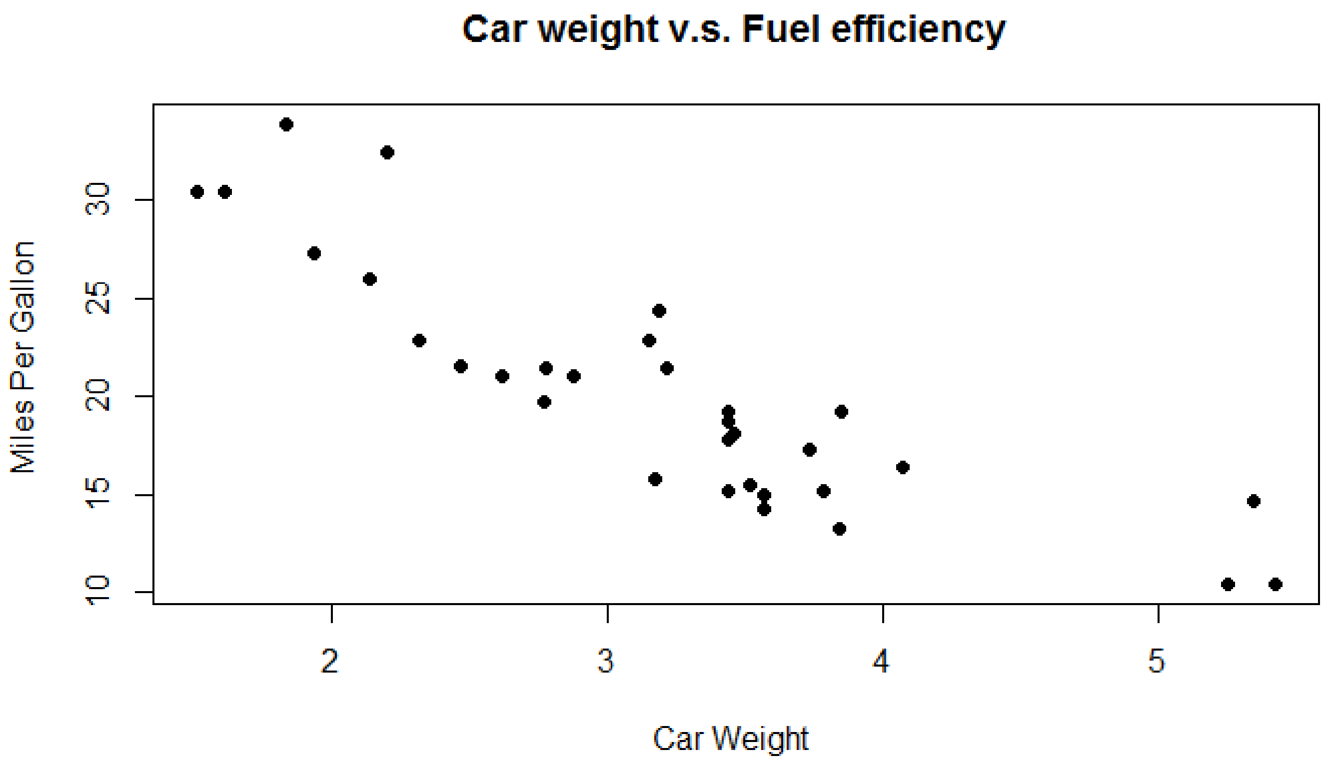
Linear Regression

- $\hat{y}_i = \beta_0 + \beta_1 x_i$
- $\varepsilon_i = y_i - (\beta_0 + \beta_1 x_i)$
- $\sum \varepsilon_i^2 = \sum\{y_i - (\beta_0 + \beta_1 x_i)\}^2 = \sum\{y_i - \hat{y}_i\}^2$



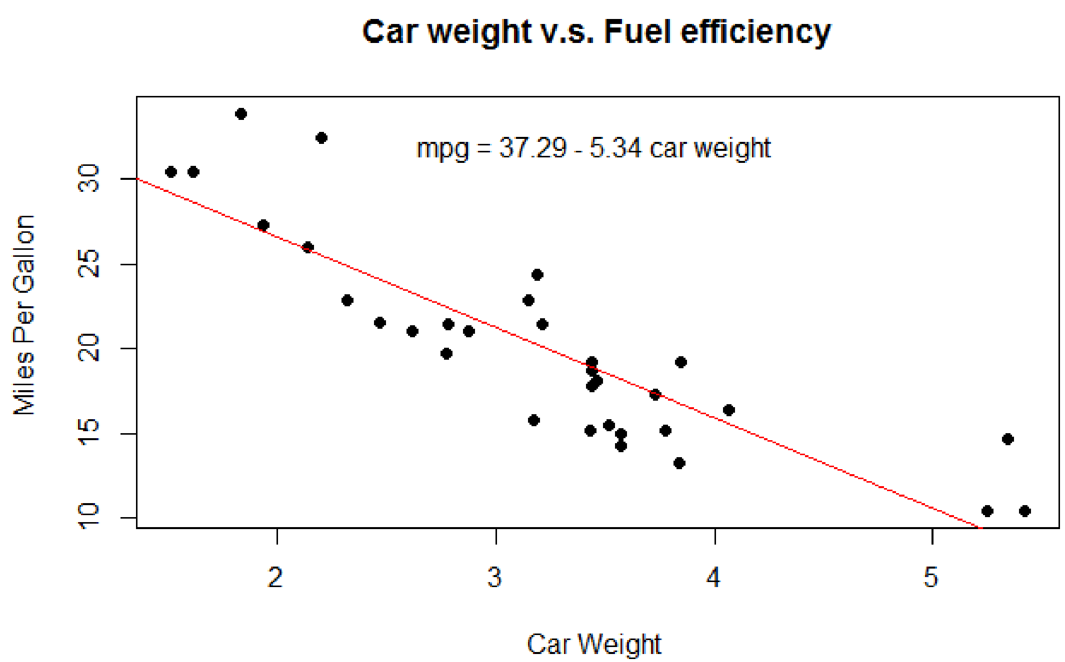
example

```
plot(mtcars$wt, mtcars$mpg,
     main = "Car weight v.s. Fuel efficiency",
     xlab = 'Car Weight', ylab = 'Miles per Gallon', pch = 19)
```



example

```
plot(mtcars$wt, mtcars$mpg,
     main = "Car weight v.s. Fuel efficiency",
     xlab = 'Car Weight', ylab = 'Miles per Gallon', pch = 19)
fit.line <- lm(mpg ~ wt, data = mtcars)
abline(fit.line, col = 'red')
cf <- round(coef(fit.line), 2)
eq <- paste0('mpg = ', cf[1],
             ifelse(sign(cf[2])==1, ' + ', ' - '),
             abs(cf[2]), " car weight ")
mtext(eq, 3, line = -2)
```

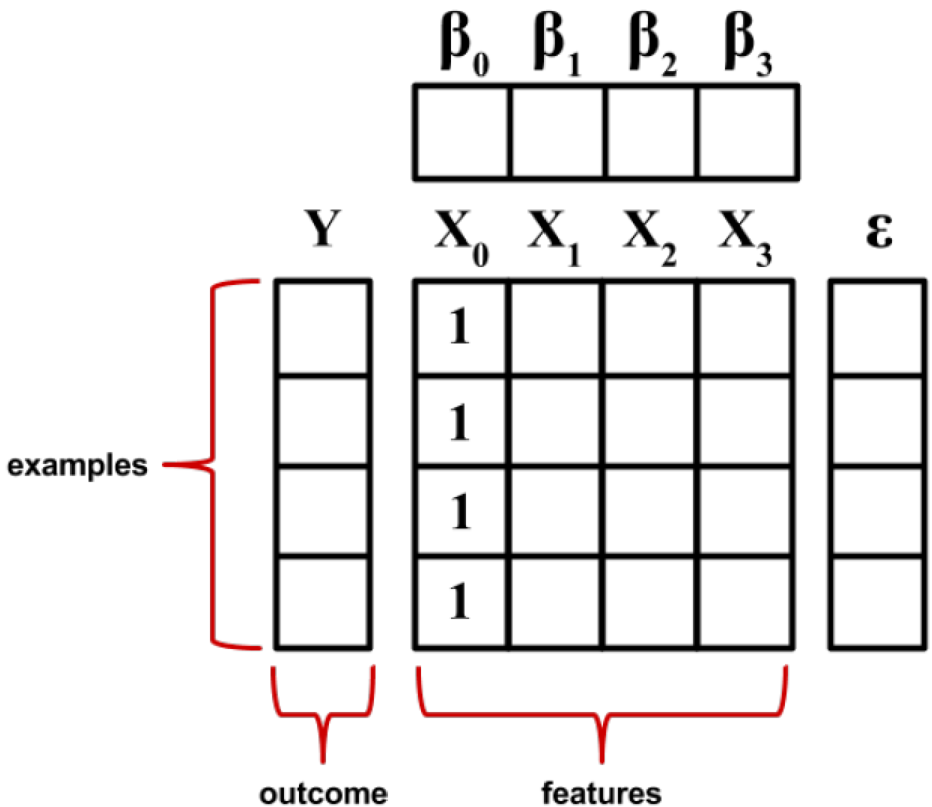


Multi-variable case

- Basic Idea
  - e.g. E(Weight Reduction) = $C_1 \times Cal.Consumption + C_2 \times$ Cal. taken

$$y = \underbrace{\alpha + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_i x_i}_{\hat{y}}$$

$$Y = X\beta + \varepsilon$$



Predicting Medical Expenses using Linear Regression

Insurance company needs to collect more in yearly premiums than it spends on medical care to its beneficiaries.

Insurance company attempts to accurately forecast medical expenses.

Medical expenses are difficult to estimate because the most costly conditions are rare and seemingly random.

Still, some conditions are more prevalent for certain segments of the population.

e.g. lung cancer is more likely among smokers than non-smokers, and heart disease may be more likely among the obese.

The goal is to use patient data to estimate the average medical care expenses.

These estimates could be used to create actuarial tables which set the price of yearly premiums higher or lower depending on the expected treatment costs.

Sample Dataset

```
load(url('https://github.com/hbchoi/SampleData/raw/master/insurance.RData'))
```

- `age`: This is an integer indicating the age of the primary beneficiary (excluding those above 64 years, since they are generally covered by the government).
- `sex`: This is the policy holder's gender, either `male` or `female`.
- `bmi`: This is the **body mass index (BMI)**, which provides a sense of how over or under-weight a person is relative to their height. BMI is equal to weight (in kilograms) divided by height (in meters) squared. An ideal BMI is within the range of 18.5 to 24.9.
- `children`: This is an integer indicating the number of children / dependents covered by the insurance plan.
- `smoker`: This is yes or no depending on whether the insured regularly smokes tobacco.
- `region`: This is the beneficiary's place of residence in the U.S., divided into four geographic regions: `northeast`, `southeast`, `southwest`, or `northwest`.

Data Exploration

```
str(insurance)
```

```
## 'data.frame':    1338 obs. of  7 variables:
##  $ age     : int  19 18 28 33 32 31 46 37 37 60 ...
##  $ sex     : Factor w/ 2 levels "female","male": 1 2 2 2 2 1 1 1 2 1 ...
##  $ bmi     : num  27.9 33.8 33 22.7 28.9 ...
##  $ children: int  0 1 3 0 0 0 1 3 2 0 ...
##  $ smoker  : Factor w/ 2 levels "no","yes": 2 1 1 1 1 1 1 1 1 1 ...
##  $ region  : Factor w/ 4 levels "northeast","northwest",..: 4 3 3 2 2 3 3 2 1 2 ...
##  $ charges : num  16885 1726 4449 21984 3867 ...
```

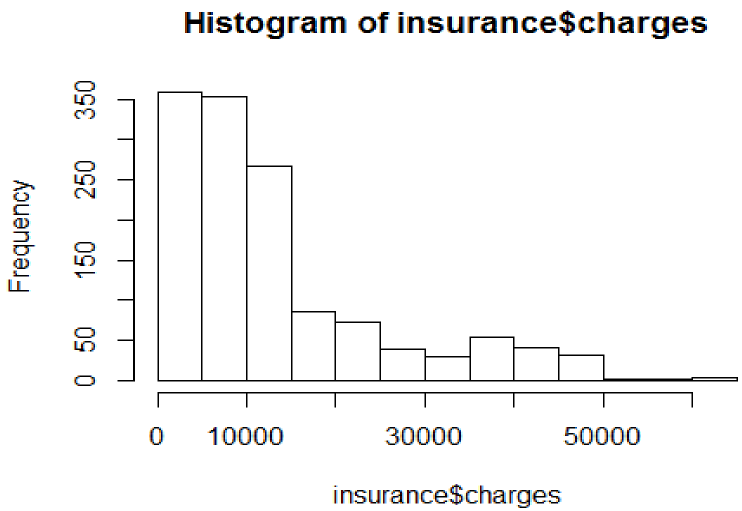
```
summary(insurance)
```

```
##       age            sex           bmi          children       smoker
##  Min.   :18.00   female:662   Min.   :15.96   Min.   :0.000   no :1064
##  1st Qu.:27.00   male  :676   1st Qu.:26.30   1st Qu.:0.000   yes: 274
##  Median :39.00                Median :30.40   Median :1.000
##  Mean   :39.21                Mean   :30.66   Mean   :1.095
##  3rd Qu.:51.00                3rd Qu.:34.69   3rd Qu.:2.000
##  Max.   :64.00                Max.   :53.13   Max.   :5.000
##        region       charges
##  northeast:324   Min.   : 1122
##  northwest:325   1st Qu.: 4740
##  southeast:364   Median : 9382
##  southwest:325   Mean   :13270
##                  3rd Qu.:16640
##                  Max.   :63770
```

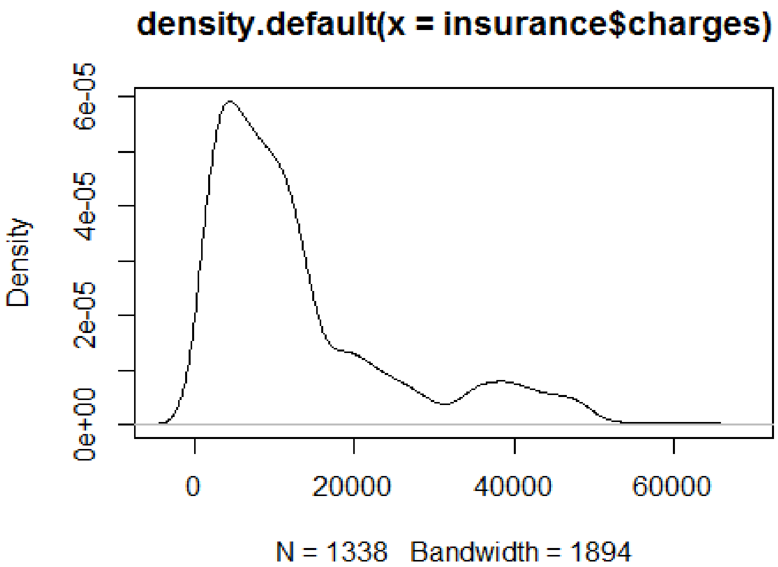
Data Exploration

- charges
  - amount of medical expenses charged by the customer – continuous value
  - regression

```
hist(insurance$charges)
```




```
plot(density(insurance$charges))
```



Data Preparation

```
set.seed(2018)
ncustomer <- nrow(insurance)
rgroup <- runif(ncustomer)

# data partition to learn a prediction model
train.df <- subset(insurance, rgroup <= 0.8)

# hold-out data for testing
test.df <- subset(insurance, rgroup > 0.8)

dim(train.df)

## [1] 1088    9

dim(test.df)

## [1] 250   9
```

- We partition the dataset into two groups with ratio of 8:2
  - train.df for building prediction model
  - test.df is to evaluate our model

Linear Regression

Multiple regression modeling syntax

using the `lm()` function in the `stats` package

Building the model:

```
m <- lm(dv ~ iv, data = mydata)
```

- `dv` is the dependent variable in the `mydata` data frame to be modeled
- `iv` is an R formula specifying the independent variables in the `mydata` data frame to use in the model
- `data` specifies the data frame in which the `dv` and `iv` variables can be found

The function will return a regression model object that can be used to make predictions. Interactions between independent variables can be specified using the * operator.

Making predictions:

```
p <- predict(m, test)
```

- `m` is a model trained by the `lm()` function
- `test` is a data frame containing test data with the same features as the training data used to build the model.

The function will return a vector of predicted values.

Example:

```
ins_model <- lm(charges ~ age + children + sex + smoker,
                          data = insurance)
ins_pred <- predict(ins_model, insurance_test)
```

Model Training

```
ins_model <- lm(charges ~ age + sex + bmi + children + smoker + region,
train.df)
ins_model
```

```
##
## Call:
## lm(formula = charges ~ age + sex + bmi + children + smoker +
##     region, data = train.df)
##
## Coefficients:
##     (Intercept)              age          sexmale              bmi
##      -11873.286          260.667            3.119          325.453
##        children        smokeryes  regionnorthwest  regionsoutheast
##         550.785        23792.886         -284.157         -852.897
## regionsouthwest
##       -1135.678
```

```
# or you could try
# lm(charges ~ . , train.df)
```

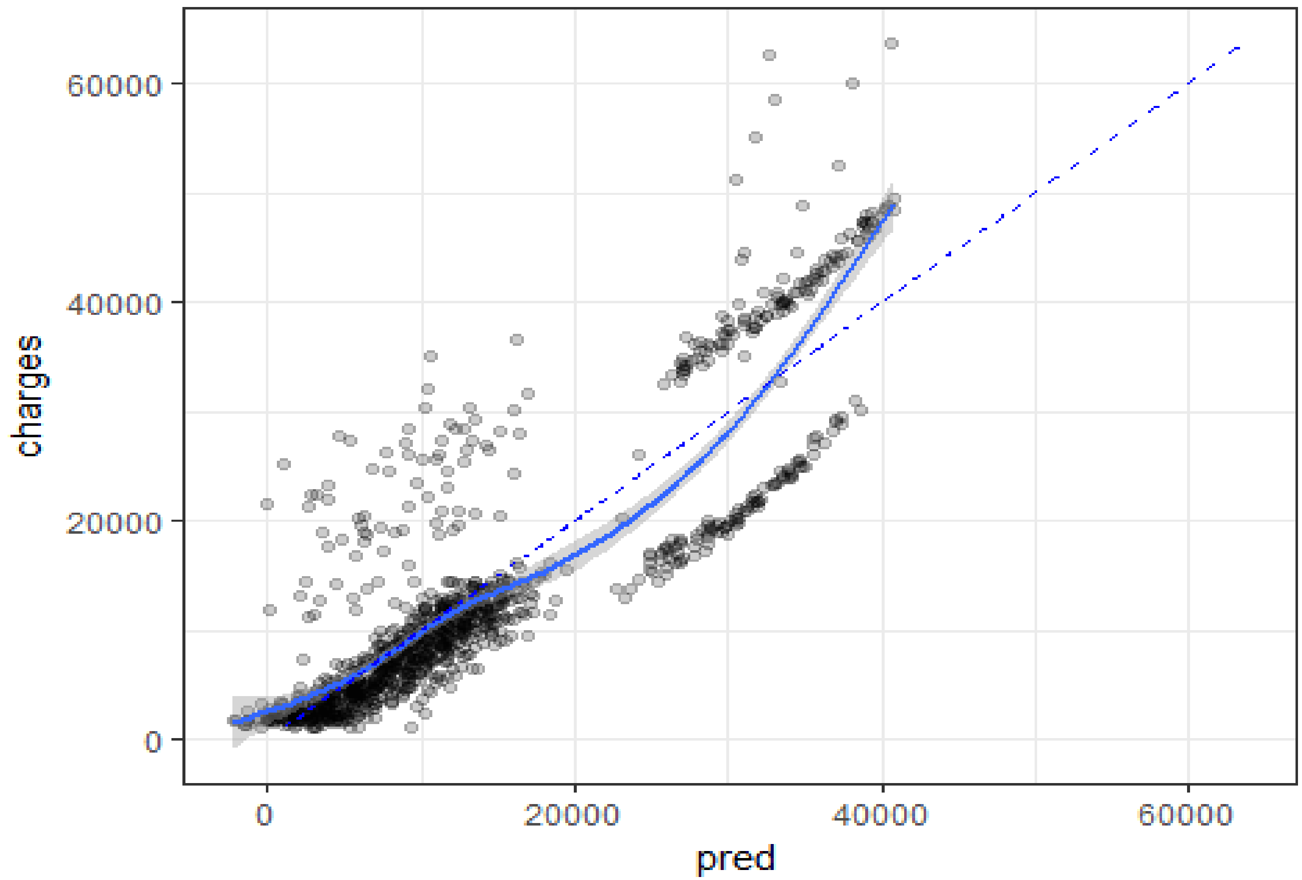
Prediction and Testing

```r
train.df$pred <- predict(ins_model, newdata = train.df)
test.df$pred <- predict(ins_model, newdata = test.df)

# performance on train.df
calcRMSE(train.df$charges, train.df$pred)

## [1] 6076.743

calcR2(train.df$charges, train.df$pred)

## [1] 0.7483358

# performance on test.df
calcRMSE(test.df$charges, test.df$pred)

## [1] 5903.393

calcR2(test.df$charges, test.df$pred)

## [1] 0.7608449
```

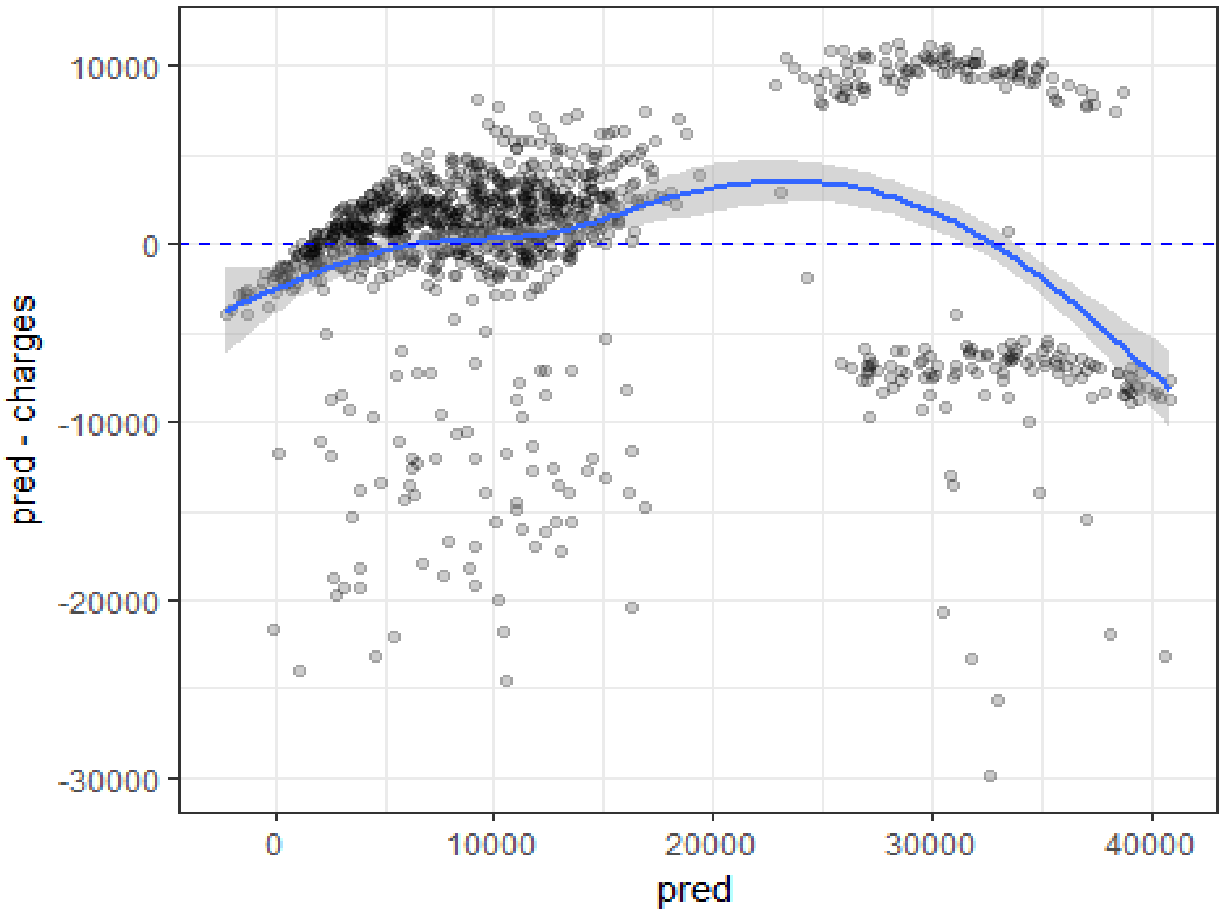
```r
calcRMSE <- function(label, estimation){
  return(sqrt(mean((label - estimation) ** 2)))
}
calcR2 <- function(label, estimation){
  RSS = sum((label - estimation) ** 2)
  SStot = sum((label - mean(label)) ** 2)

  return(1-RSS/SStot)
}
```

```
summary(ins_model)

##
## Call:
## lm(formula = charges ~ age + sex + bmi + children + smoker +
##     region, data = train.df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11311.7  -2767.7   -985.7   1322.0  29912.3
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -11873.286   1114.946 -10.649  < 2e-16 ***
## age                260.667     13.333  19.551  < 2e-16 ***
## sexmale              3.119    371.576   0.008 0.993304
## bmi                325.453     32.285  10.081  < 2e-16 ***
## children           550.785    154.311   3.569 0.000374 ***
## smokeryes        23792.886    460.478  51.670  < 2e-16 ***
## regionnorthwest   -284.157    529.943  -0.536 0.591928
## regionsoutheast   -852.897    534.558  -1.596 0.110890
## regionsouthwest  -1135.678    533.061  -2.130 0.033357 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6102 on 1079 degrees of freedom
## Multiple R-squared:  0.7483, Adjusted R-squared:  0.7465
## F-statistic: 401.1 on 8 and 1079 DF,  p-value: < 2.2e-16
```

Visualization

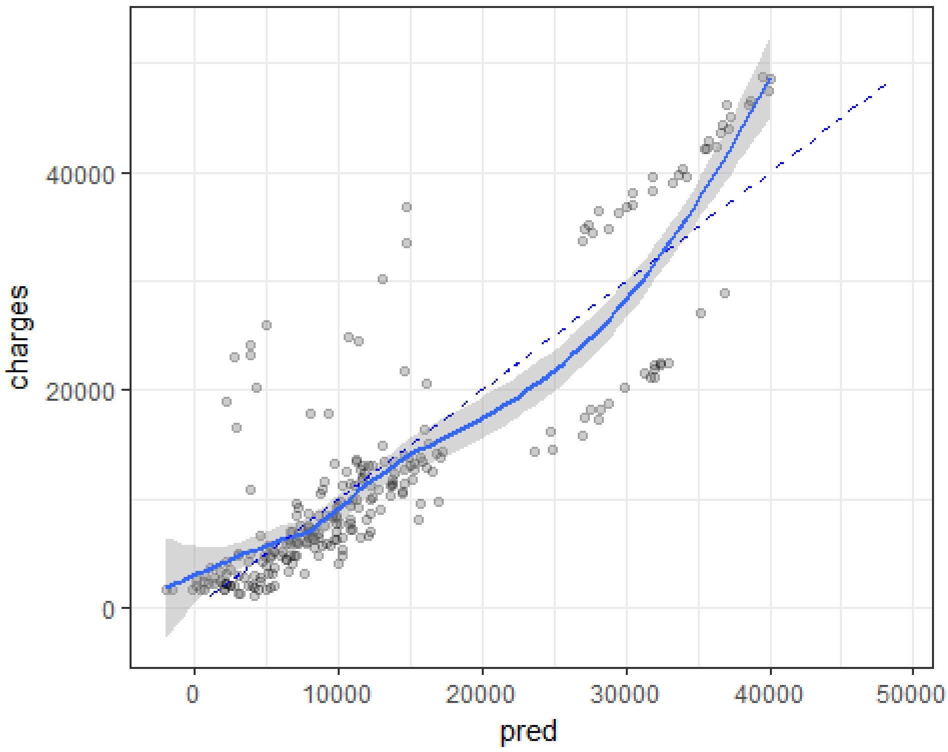


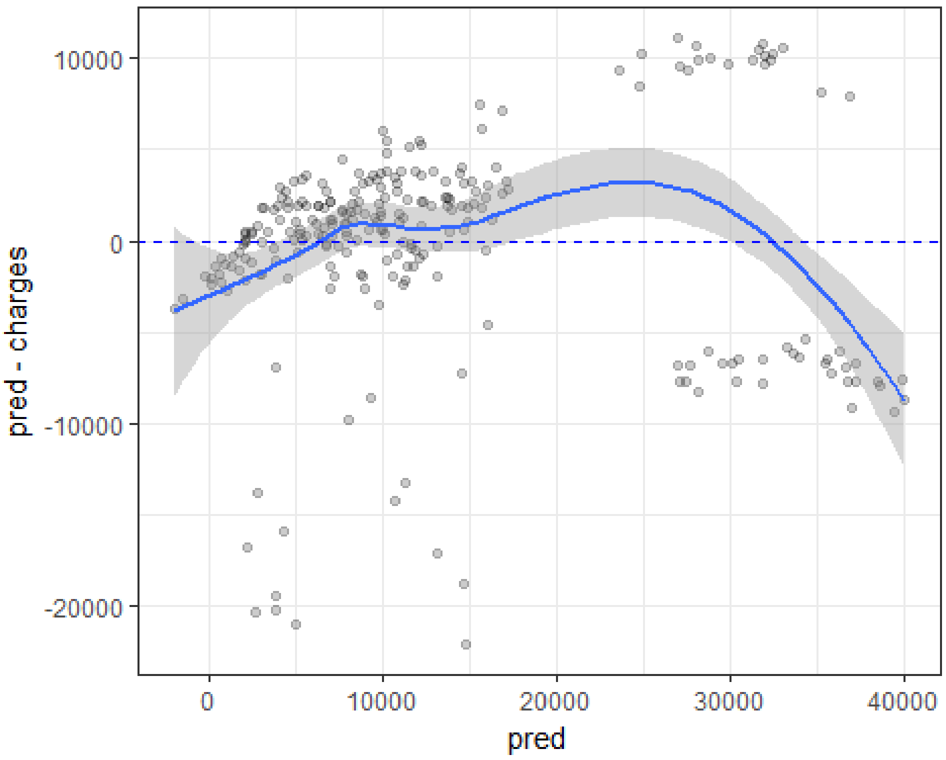

```
library(ggplot2)

ggplot(train.df, aes(x = pred, y = charges)) +
  geom_point(alpha = 0.2, col = 'black') +
  geom_smooth()+
  geom_line(aes(x = charges, y = charges), col = 'blue', linetype = 2)
```




```
ggplot(train.df, aes(x = pred, y = pred - charges)) +
  geom_point(alpha = 0.2, col = 'black') +
  geom_smooth()+
  geom_hline(yintercept = 0, col = 'blue', linetype = 2)
```

Test dataset





Improvement

Adding non-linear relationships

e.g. `charges ~ age + I(age ^ 2)`

Converting Num. var. into a binary indicator (or range variable)

e.g. `insurance$bmi30 <- ifelse(insurance$bmi >= 30, 1, 0)`

Improvement

Adding Interaction

The simultaneous influence of two variables on the outcome is not additive

```
plant_height ~ bacteria + sun + bacteria:sun
```

or equivalently

```
plant_height ~ bacteria*sun
```

- ✓ Change in height is more (or less) than the sum of the effects due to sun/bacteria
- ✓ At higher levels of sunlight, 1 unit change in bacteria causes more change in height

```r
train.df$bmi30 <- ifelse(train.df$bmi >= 30, 1, 0)
test.df$bmi30 <- ifelse(test.df$bmi >= 30, 1, 0)

ins_model <- lm(charges ~ age + I(age^2) + sex + bmi + children + bmi30
* smoker + region, train.df)
ins_model
```

```
##
## Call:
## lm(formula = charges ~ age + I(age^2) + sex + bmi + children +
##     bmi30 * smoker + region, data = train.df)
##
## Coefficients:
##     (Intercept)               age          I(age^2)           sexmale
##        -613.136            24.543             3.099          -297.927
##             bmi          children             bmi30         smokeryes
##         105.989           672.386         -1150.657         13301.273
## regionnorthwest  regionsoutheast  regionsouthwest  bmi30:smokeryes
##        -309.922          -614.716         -1394.706         20175.516
```

```
summary(ins_model)
```

```
##
## Call:
## lm(formula = charges ~ age + I(age^2) + sex + bmi + children +
##     bmi30 * smoker + region, data = train.df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17553.0  -1706.4  -1246.2   -670.6  24255.5
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -613.1359  1533.2402  -0.400 0.689313
## age                24.5432    66.9946   0.366 0.714179
## I(age^2)            3.0987     0.8377   3.699 0.000227 ***
## sexmale          -297.9265   270.2113  -1.103 0.270461
## bmi               105.9893    38.0656   2.784 0.005457 **
## children          672.3862   117.5630   5.719 1.38e-08 ***
## bmi30           -1150.6569   465.0201  -2.474 0.013498 *
## smokeryes       13301.2726   481.9281  27.600  < 2e-16 ***
## regionnorthwest  -309.9223   385.1832  -0.805 0.421223
## regionsoutheast  -614.7161   389.1414  -1.580 0.114476
## regionsouthwest -1394.7060   387.4405  -3.600 0.000333 ***
## bmi30:smokeryes 20175.5159   666.8048  30.257  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4434 on 1076 degrees of freedom
## Multiple R-squared:  0.8675, Adjusted R-squared:  0.8661
## F-statistic: 640.3 on 11 and 1076 DF,  p-value: < 2.2e-16
```

```r
train.df$pred <- predict(ins_model, newdata = train.df)
test.df$pred <- predict(ins_model, newdata = test.df)

# performance on train.df
calcRMSE(train.df$charges, train.df$pred)
```

```
## [1] 4409.541
```

```r
calcR2(train.df$charges, train.df$pred)
```

```
## [1] 0.8674846
```

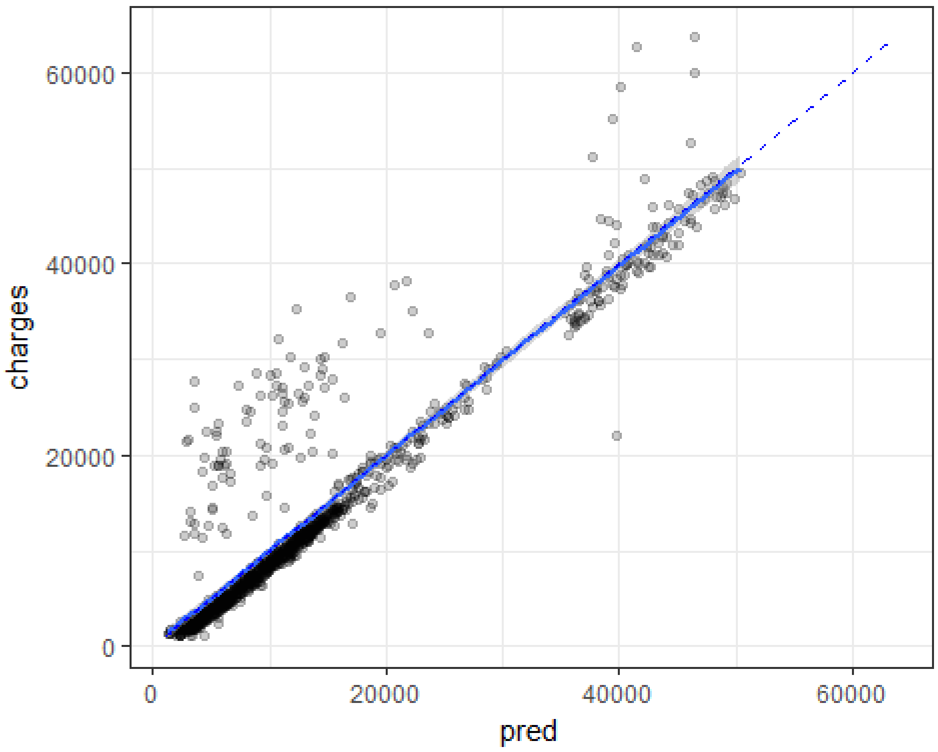
```r
# performance on test.df
calcRMSE(test.df$charges, test.df$pred)
```

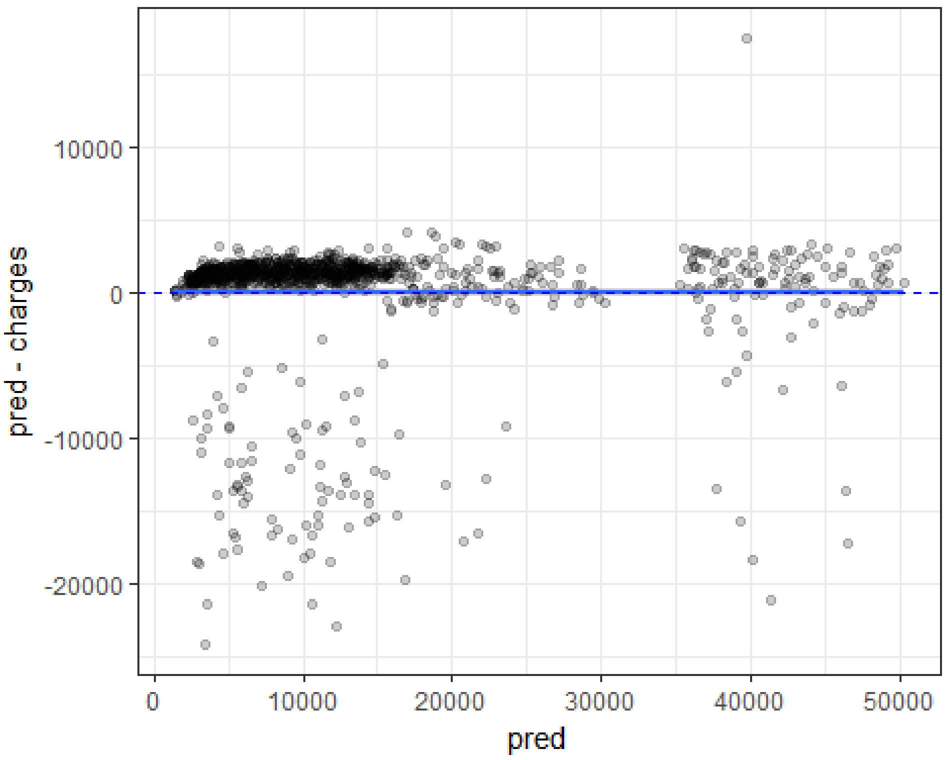
```
## [1] 4528.097
```

```r
calcR2(test.df$charges, test.df$pred)
```

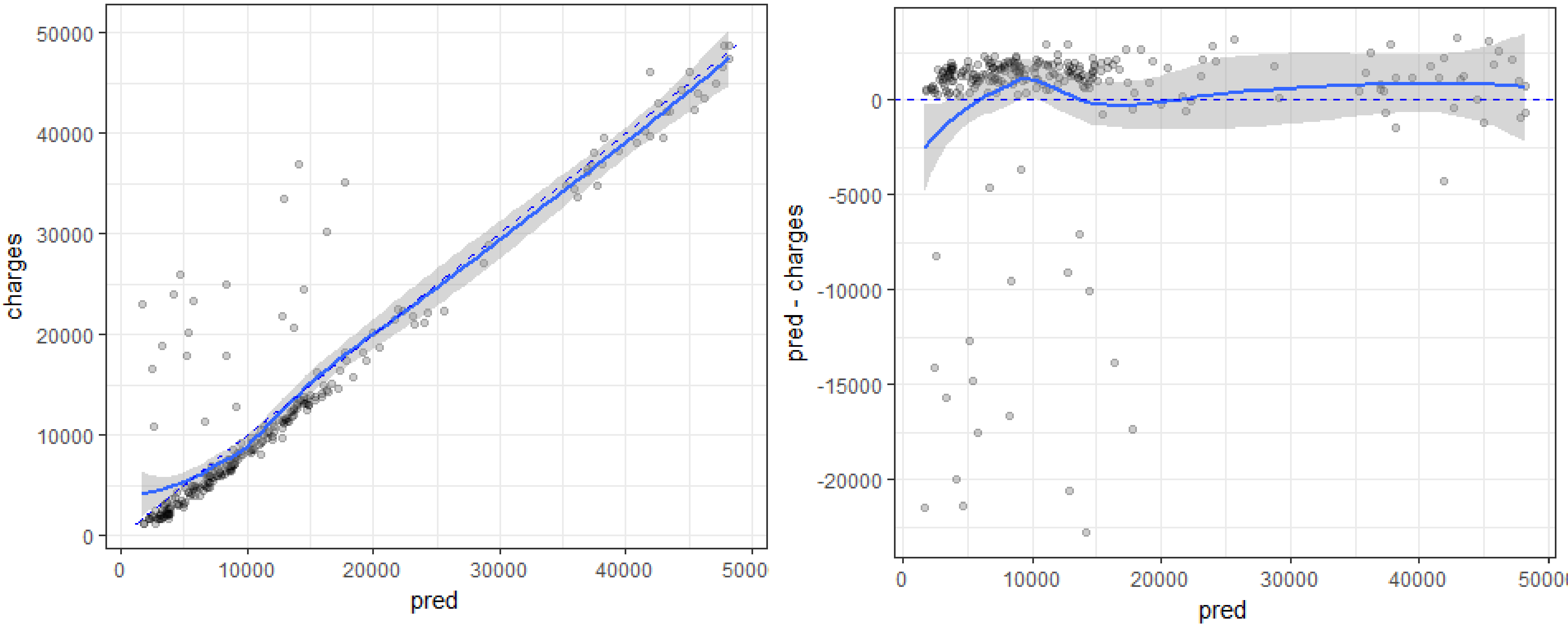
```
## [1] 0.8592956
```

Visualization on Training Data





Visualization on Test Data



References

- Practical Data Science with R, by Nina Zumel and John Mount
- R을 이용한 데이터 분석 실무, 서민구, 길벗
- Machine Learning with R, by Brett Lantz
  - 한글판, R을 활용한 기계 학습
- [DBGUIDE 연재] ggplot2를 이용한 R 시각화
  - http://freesearch.pe.kr/archives/3134