Multiple Linear Regression

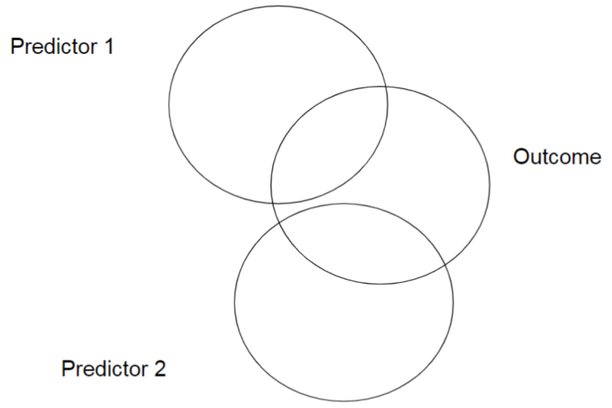
John Rios

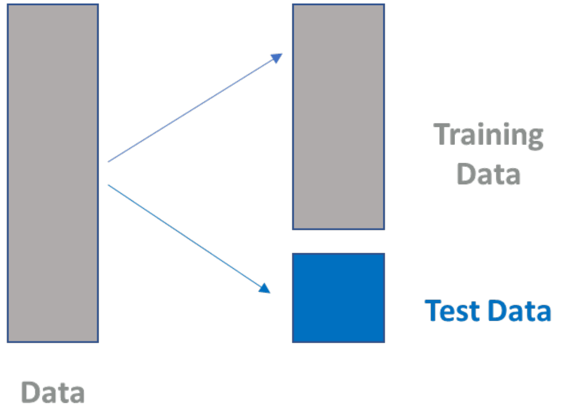
Business Intelligence

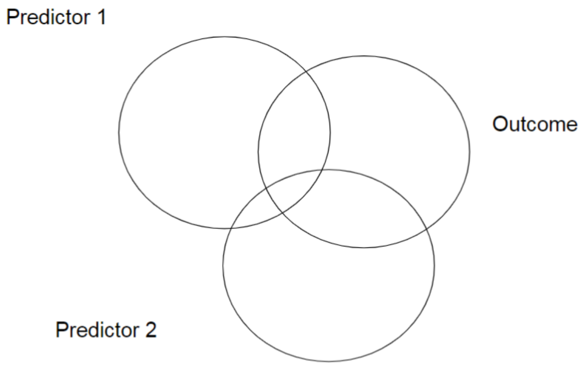
Terry College of Business
UNIVERSITY OF GEORGIA

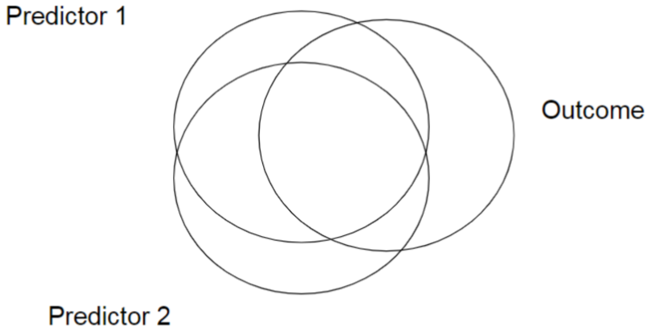
$$y = \alpha + \beta 1x1 + \beta 2x2 + \ldots + \beta kxk$$

y = target (or dependent variable)
α = y-intercept
β = coefficients assigned to each of the IVs
x = predictors (or independent variables)









Multiple Linear Regression

- In most prediction situations, there are a variety of predictors to be used
- What you want is an equation that represents the relationship between the outcome variable and the set of predictors

Multiple regression model

- Multiple regression is an extension of simple linear regression in which several IVs, instead of just one, are combined to predict a value on a DV for each case.

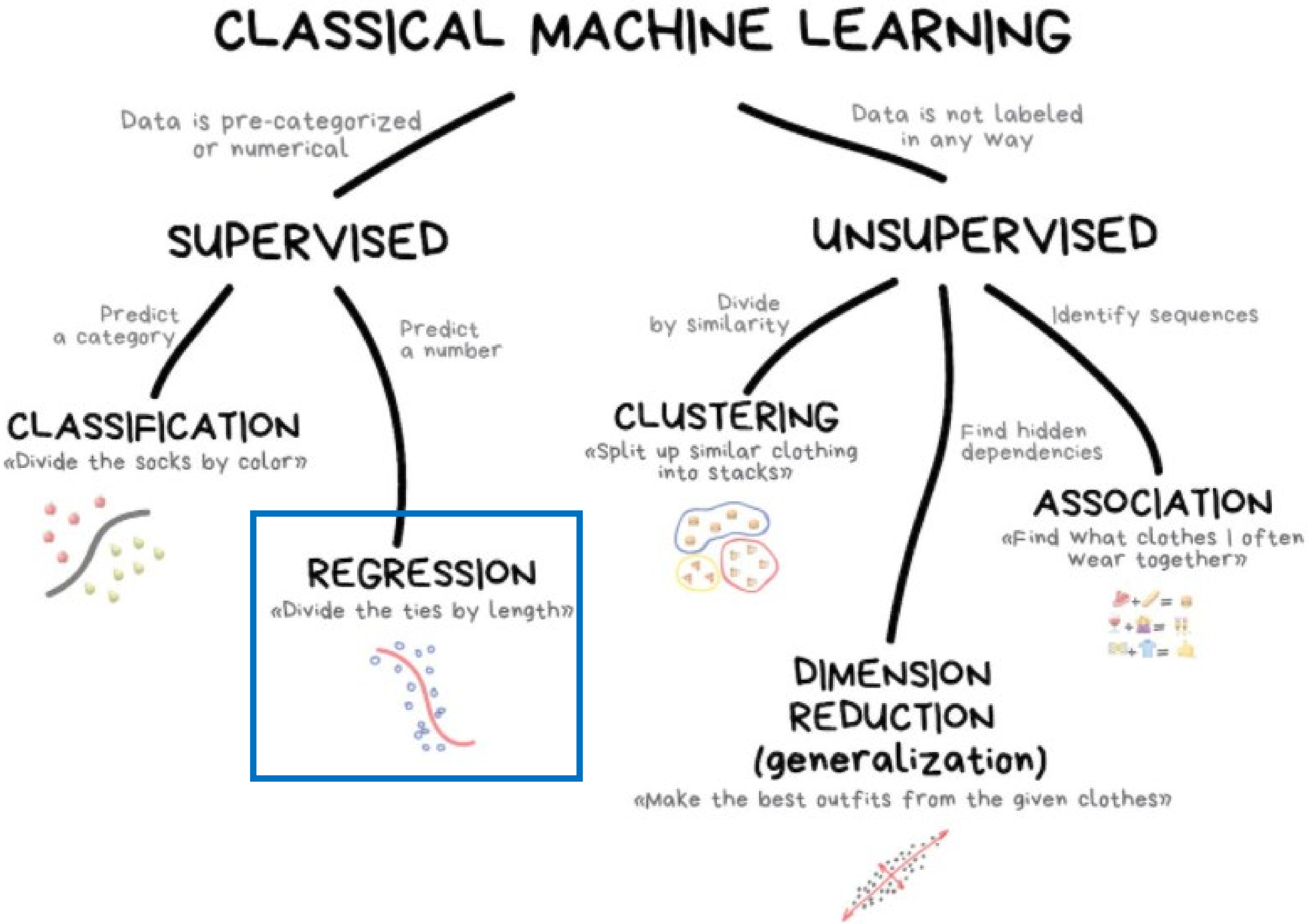
$$\boldsymbol{y} = \boldsymbol{\alpha} + \boldsymbol{\beta}_1 \boldsymbol{x}_1 + \boldsymbol{\beta}_2 \boldsymbol{x}_2 + \ldots + \boldsymbol{\beta}_k \boldsymbol{x}_k$$

- y = target (or dependent variable)
- α = y-intercept
- β = coefficients assigned to each of the IVs
- x = predictors (or independent variables)

Limitations to multiple regression analyses

- Regression analyses reveal relationships among variables but do not imply that the relationships are causal.
- Inclusion of variables: Which DV should be used, and how is it to be measured? Which IVs should be examined, and how are they to be measured?
- A multiple regression solution is extremely sensitive to the combination of variables that is included in it.
- Extreme cases have too much impact on the regression solution and affect the precision of estimation of the regression weights.

Tabacknick et al (2019). Using Multivariate Statistics.


CLASSICAL MACHINE LEARNING
Data is pre-categorized
or numerical
Data is not labeled
in any way
SUPERVISED
UNSUPERVISED
Predict
a category
Predict
a number
Divide
by similarity
Identify sequences
CLASSIFICATION
«Divide the socks by color»
REGRESSION
«Divide the ties by length»
CLUSTERING
«Split up similar clothing
into stacks»
Find hidden
dependencies
ASSOCIATION
«Find what clothes I often
wear together»
DIMENSION
REDUCTION
(generalization)
«Make the best outfits from the given clothes»

Regression Analysis – Two Approaches

- Prediction (Machine Learning)

Predict values of the outcome variable from values of the predictor variable

- Explanation

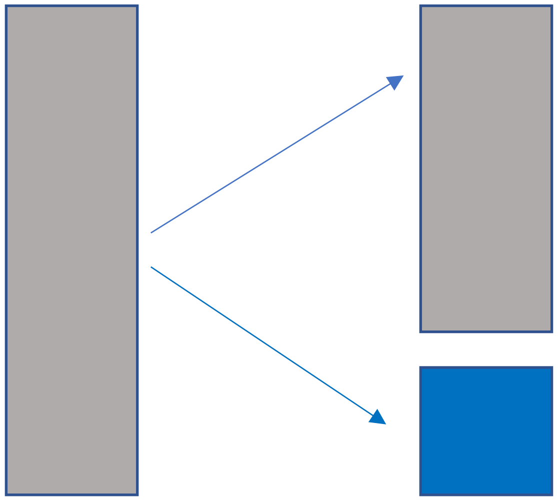
Determine the amount of variance in the outcome variable that is explained by the predictor variable(s)

Determine which predictors are the most useful for predicting the outcome variable

Prediction (ML) Approach

Machine Learning Use

- Predictive modeling
- Evaluate based on prediction error



Model Evaluation

How well the model predicts new data (*not* how well it fits the data it was trained with)

- Key component of most measures is difference between actual outcome and predicted outcome (i.e., error)

Model Evaluation

Error for data record = predicted (p) minus actual (a)

RMSE: Root Mean Squared Error: $\sqrt{\frac{1}{n}\sum_1^n\left(Y_i - \hat{Y}_i\right)^2}$

MAE: Mean Absolute Error: $\frac{1}{n}\sum_1^n\left|\left(Y_i - \hat{Y}_i\right)\right|$

MAPE: Mean Absolute Percentage Error: $\frac{100}{n}\sum_1^n\left|\frac{Y_i - \hat{Y}_i}{Y_i}\right|$

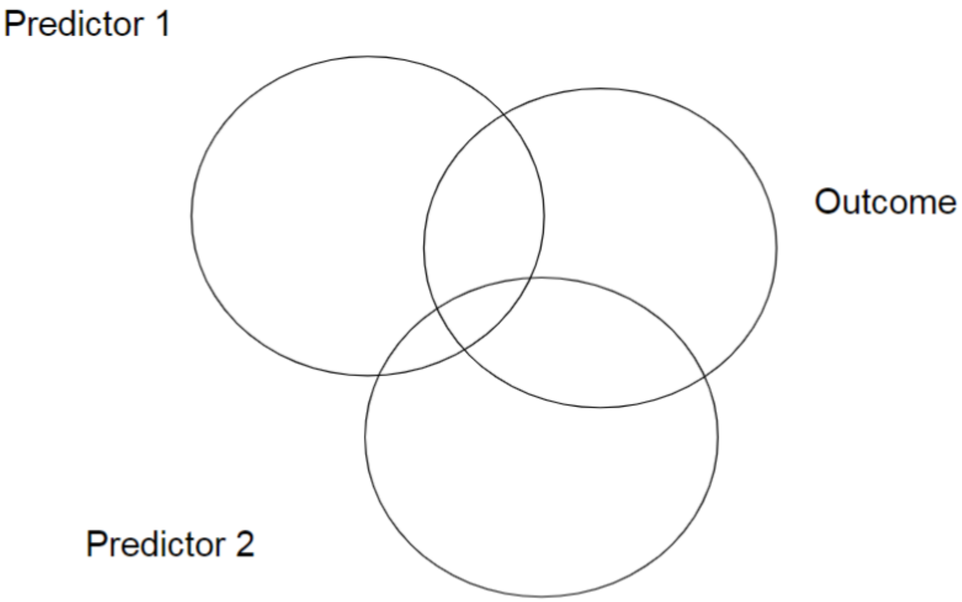
Total SSE: Total Sum of Squared Errors: $\sum_1^n\left(Y_i - \hat{Y}_i\right)^2$

Explanation Approach

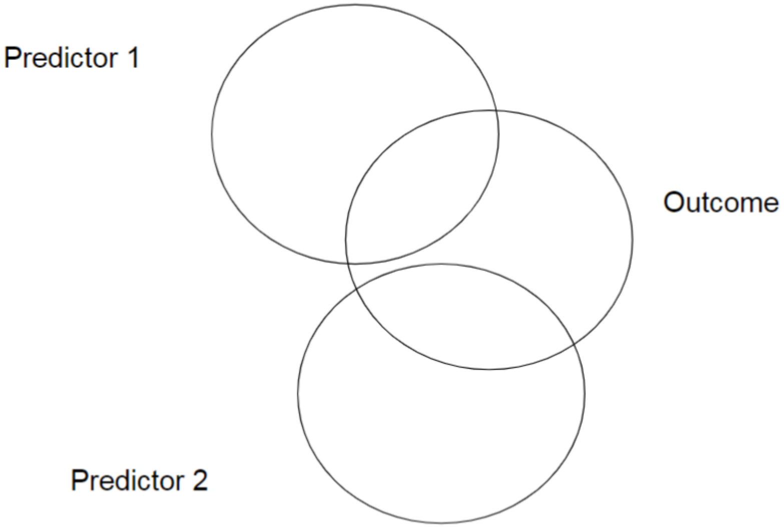
Multicollinearity

- Fancy term for “correlated predictors”
- Makes interpretation of weights difficult
- When two predictors are strongly related to one another, one of the predictors receives a large weight in the proper direction, while the other receives a small or counterintuitive weight (sometimes in the wrong direction)

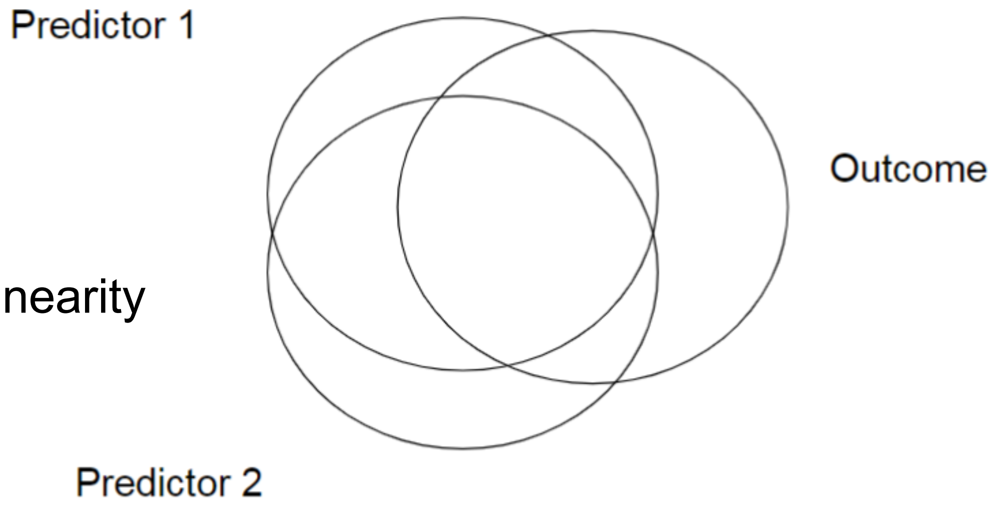
Explanation - Multicollinearity



Minimal Multicollinearity



Zero Multicollinearity

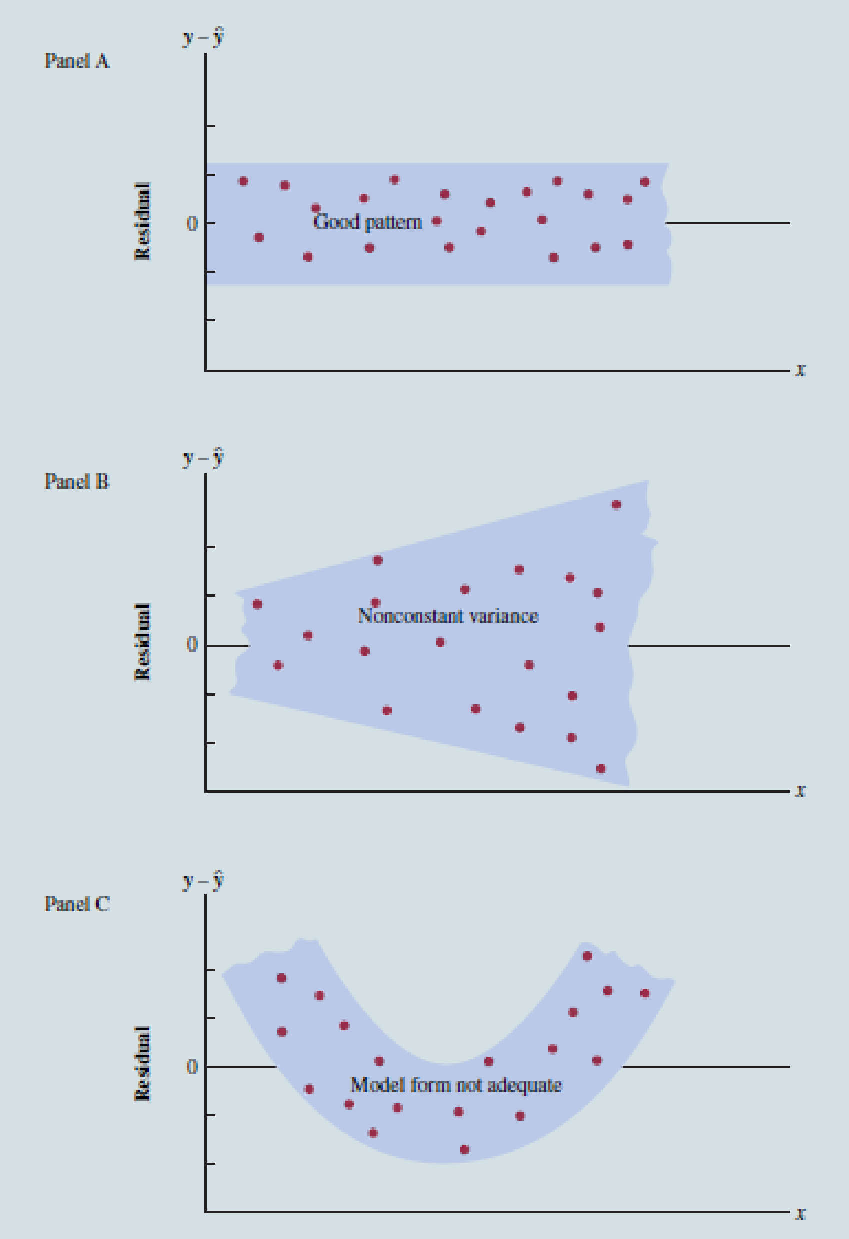


High Multicollinearity

Explanation - Reg. Assumptions

- Linear relationships
- Normally distributed errors (no pattern)
  - Independent
  - Similar variance across range of X
  - Eyeball test of plots

FIGURE 14.12 Residual Plots from Three Regression Studies



Multiple Linear Regression in R

Explanation Approach

Multiple Linear Regression in R

Use the the **`caret`** package

Insurance dataset – recall, *the goal is to explain the target as best as we can*

```
library(tidyverse)
library(caret)

insurance <- read_csv("insurance.csv")
```

Selecting Predictors

To compute the correlation, we need numeric values

```
# transform categories to numbers
library(fastDummies)
insurance <- insurance %>%
  mutate(sexN = case_when(
    sex == "male" ~ 1,
    sex == "female" ~ 0
    )) %>%
  mutate(smokerN = case_when(
    smoker == "yes" ~ 1,
    smoker == "no" ~ 0
    )) %>%
  dummy_cols(., select_columns =
                'region')
```

```
# only select numeric variables
df <- insurance %>%
  dplyr::select(charges, age, sexN, bmi,
children, smokerN, region_northeast,
region_northwest, region_southeast,
region_southwest)


# drop missing values NAs
df1 <- drop_na(df)
```

Multicollinearity Check

```
# compute correlation between predictors
cor(df1[,2:10])
```

```
> cor(df1[,2:10])
                           age         sexN          bmi    children      smokerN region_northeast region_northwest
age               1.0000000000 -0.020855872  0.109271882  0.04246900 -0.025018752      0.002474955    -0.0004074234
sexN             -0.0208558722  1.000000000  0.046371151  0.01716298  0.076184817     -0.002425432    -0.0111557280
bmi               0.1092718815  0.046371151  1.000000000  0.01275890  0.003750426     -0.138156224    -0.1359955237
children          0.0424689986  0.017162978  0.012758901  1.00000000  0.007673120     -0.022807598     0.0248061293
smokerN          -0.0250187515  0.076184817  0.003750426  0.00767312  1.000000000      0.002811135    -0.0369454740
region_northeast  0.0024749545 -0.002425432 -0.138156224 -0.02280760  0.002811135      1.000000000    -0.3201772613
region_northwest -0.0004074234 -0.011155728 -0.135995524  0.02480613 -0.036945474     -0.320177261     1.0000000000
region_southeast -0.0116419406  0.017116875  0.270024649 -0.02306575  0.068498410     -0.345561015    -0.3462646614
region_southwest  0.0100162342 -0.004184049 -0.006205183  0.02191358 -0.036945474     -0.320177261    -0.3208292201
                 region_southeast region_southwest
age                   -0.01164194      0.010016234
sexN                   0.01711688     -0.004184049
bmi                    0.27002465     -0.006205183
children              -0.02306575      0.021913576
smokerN                0.06849841     -0.036945474
region_northeast      -0.34556102     -0.320177261
region_northwest      -0.34626466     -0.320829220
region_southeast       1.00000000     -0.346264661
region_southwest      -0.34626466      1.000000000
```

Exclude a Dummy

```
# compute correlation between predictors and the target
cor(df1[,1:10])
```

```
> cor(df1[,1:10])
                      charges           age         sexN          bmi    children      smokerN region_northeast
charges           1.000000000  0.2990081933  0.057292062  0.198340969  0.06799823  0.787251430      0.006348771
age               0.299008193  1.0000000000 -0.020855872  0.109271882  0.04246900 -0.025018752      0.002474955
sexN              0.057292062 -0.0208558722  1.000000000  0.046371151  0.01716298  0.076184817     -0.002425432
bmi               0.198340969  0.1092718815  0.046371151  1.000000000  0.01275890  0.003750426     -0.138156224
children          0.067998227  0.0424689986  0.017162978  0.012758901  1.00000000  0.007673120     -0.022807598
smokerN           0.787251430 -0.0250187515  0.076184817  0.003750426  0.00767312  1.000000000      0.002811135
region_northeast  0.006348771  0.0024749545 -0.002425432 -0.138156224 -0.02280760  0.002811135      1.000000000
region_northwest -0.039904864 -0.0004074234 -0.011155728 -0.135995524  0.02480613 -0.036945474     -0.320177261
region_southeast  0.073981552 -0.0116419406  0.017116875  0.270024649 -0.02306575  0.068498410     -0.345561015
region_southwest -0.043210029  0.0100162342 -0.004184049 -0.006205183  0.02191358 -0.036945474     -0.320177261
                 region_northwest region_southeast region_southwest
charges             -0.0399048640       0.07398155     -0.043210029
age                 -0.0004074234      -0.01164194      0.010016234
sexN                -0.0111557280       0.01711688     -0.004184049
bmi                 -0.1359955237       0.27002465     -0.006205183
children             0.0248061293      -0.02306575      0.021913576
smokerN             -0.0369454740       0.06849841     -0.036945474
region_northeast    -0.3201772613      -0.34556102     -0.320177261
region_northwest     1.0000000000      -0.34626466     -0.320829220
region_southeast    -0.3462646614       1.00000000     -0.346264661
region_southwest    -0.3208292201      -0.34626466      1.000000000
```

Model Induction

```
# run the model with the entire dataset and all the features (be aware of dummies)
df2 <- df1 %>%
  dplyr::select(charges, age, sexN, bmi, children, smokerN,
                region_northwest, region_southeast, region_southwest)

model <- train(charges ~ .,
               data = df2, # data set
               method = "lm") # linear regression
```

Model Performance

```
# check the results
summary(model)
```

```
Residual standard error: 6062 on 1329 degrees of freedom
Multiple R-squared:  0.7509,	Adjusted R-squared:  0.7494
F-statistic: 500.8 on 8 and 1329 DF,  p-value: < 2.2e-16

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       -11938.5      987.8 -12.086  < 2e-16 ***
age                  256.9       11.9  21.587  < 2e-16 ***
sexN                -131.3      332.9  -0.394 0.693348
bmi                  339.2       28.6  11.860  < 2e-16 ***
children             475.5      137.8   3.451 0.000577 ***
smokerN            23848.5      413.1  57.723  < 2e-16 ***
region_northwest    -353.0      476.3  -0.741 0.458769
region_southeast   -1035.0      478.7  -2.162 0.030782 *
region_southwest    -960.0      477.9  -2.009 0.044765 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Selecting Predictors

```
# compute correlation between predictors and the target
cor(df1[,1:10])
```

```
> cor(df1[,1:10])
                      charges           age          sexN          bmi   children      smokerN region_northeast
charges           1.000000000  0.2990081933  0.057292062  0.198340969 0.06799823  0.787251430      0.006348771
age               0.299008193  1.0000000000 -0.020855872  0.109271882 0.04246900 -0.025018752      0.002474955
sexN              0.057292062 -0.0208558722  1.000000000  0.046371151 0.01716298  0.076184817     -0.002425432
bmi               0.198340969  0.1092718815  0.046371151  1.000000000 0.01275890  0.003750426     -0.138156224
children          0.067998227  0.0424689986  0.017162978  0.012758901 1.00000000  0.007673120     -0.022807598
smokerN           0.787251430 -0.0250187515  0.076184817  0.003750426 0.00767312  1.000000000      0.002811135
region_northeast  0.006348771  0.0024749545 -0.002425432 -0.138156224 -0.02280760  0.002811135      1.000000000
region_northwest -0.039904864 -0.0004074234 -0.011155728 -0.135995524 0.02480613 -0.036945474     -0.320177261
region_southeast  0.073981552 -0.0116419406  0.017116875  0.270024649 -0.02306575  0.068498410     -0.345561015
region_southwest -0.043210029  0.0100162342 -0.004184049 -0.006205183 0.02191358 -0.036945474     -0.320177261
                 region_northwest region_southeast region_southwest
charges             -0.0399048640       0.07398155     -0.043210029
age                 -0.0004074234      -0.01164194      0.010016234
sexN                -0.0111557280       0.01711688     -0.004184049
bmi                 -0.1359955237       0.27002465     -0.006205183
children             0.0248061293      -0.02306575      0.021913576
smokerN             -0.0369454740       0.06849841     -0.036945474
region_northeast    -0.3201772613      -0.34556102     -0.320177261
region_northwest     1.0000000000      -0.34626466     -0.320829220
region_southeast    -0.3462646614       1.00000000     -0.346264661
region_southwest    -0.3208292201      -0.34626466      1.000000000
```

Model Induction

```
# run the model with the entire dataset and the selected features
model <- train(charges ~ age + bmi + smokerN + region_southeast,
               data = df2, # data set
               method = "lm") # linear regression
```

Model Performance

```
# check the results
summary(model)
```

```
Residual standard error: 6089 on 1333 degrees of freedom
Multiple R-squared:  0.7479,	Adjusted R-squared:  0.7472 
F-statistic: 988.9 on 4 and 1333 DF,  p-value: < 2.2e-16

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)    
(Intercept)      -11865.27     944.66 -12.560   <2e-16 ***
age                 258.77      11.94  21.677   <2e-16 ***
bmi                 334.90      28.56  11.727   <2e-16 ***
smokerN           23868.68     413.63  57.706   <2e-16 ***
region_southeast   -613.79     389.78  -1.575    0.116    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

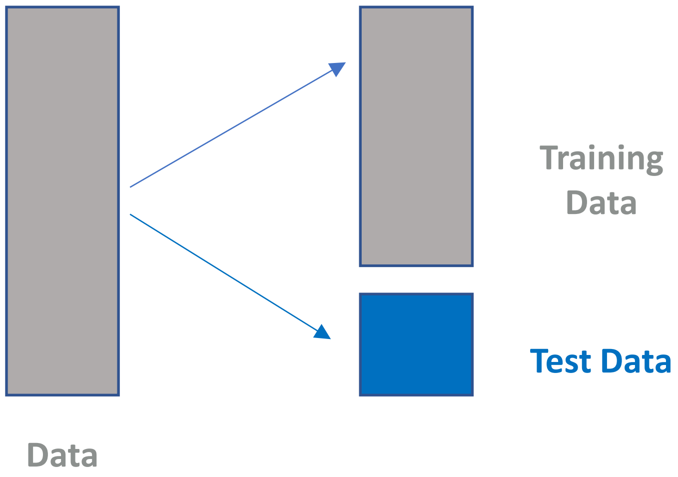
Multiple Linear Regression in R

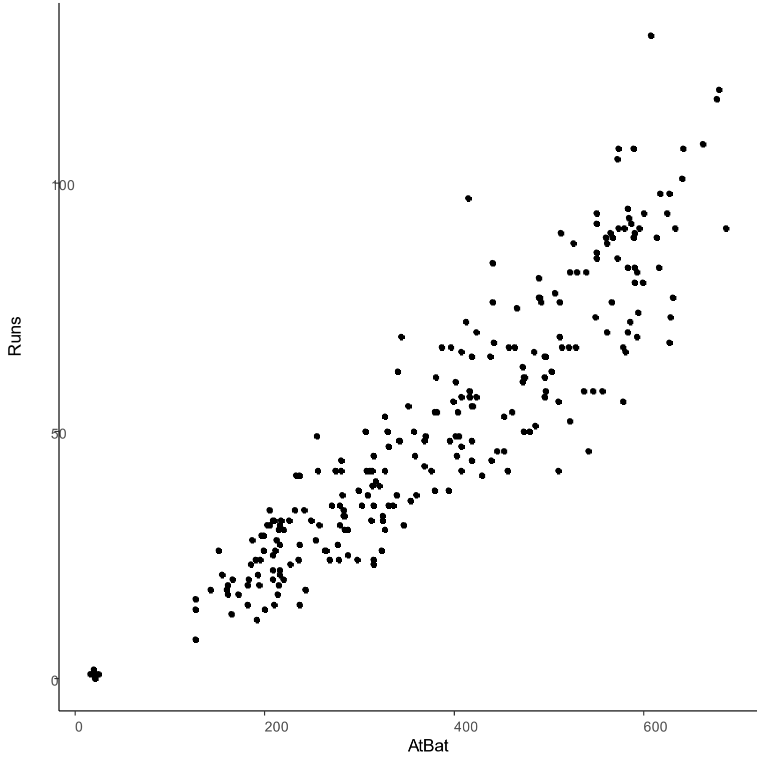
Prediction (ML) Approach

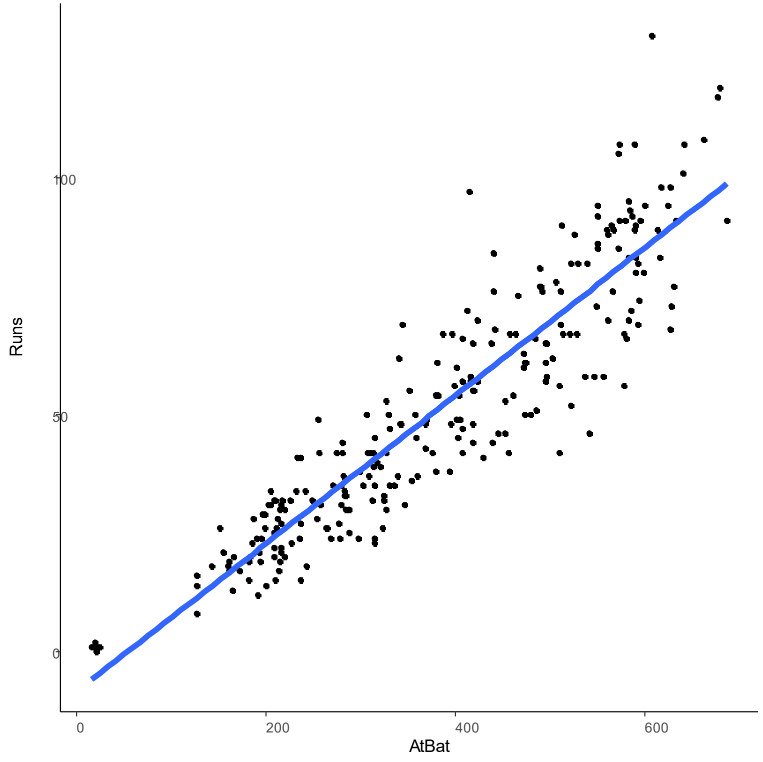
Multiple Linear Regression - Prediction

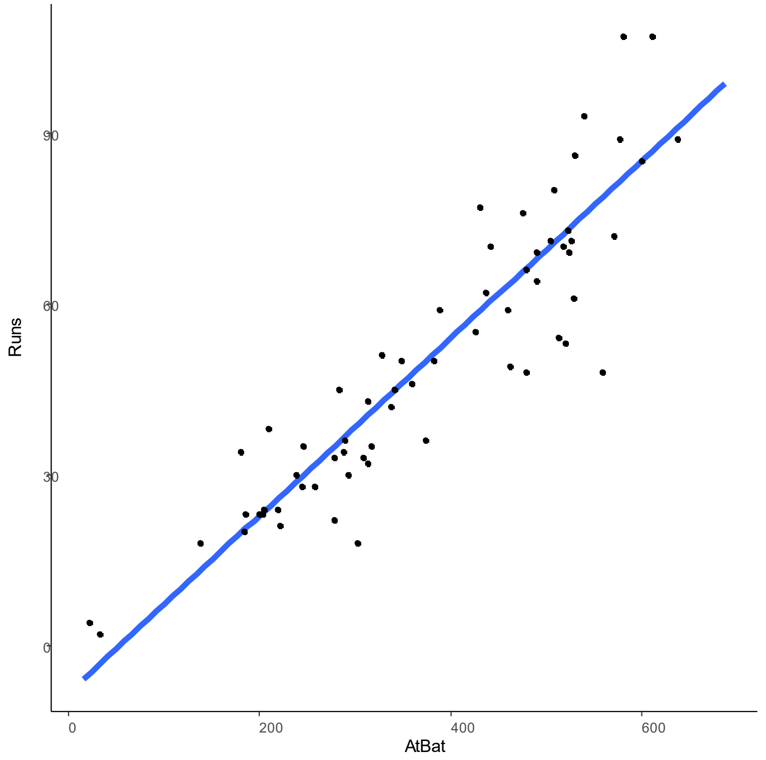
The goal is to predict the target using a new dataset as best as we can

Train and Test the Model









Splitting Data

Set a starting value (seed) so that results are reproducible
Split the data into training and testing

```
set.seed(12L) # set a starting seed to be able to get reproducible results

# partition data
trainIndex <- createDataPartition(df1$charges, # target variable
                                  p = 0.8, # percentage that goes to training
                                  list = FALSE, # results will not be in a list
                                  times = 1) # number of partitions to create

charges_train <- df1[trainIndex, ] # tibble (data frame) for training
charges_test <- df1[-trainIndex, ] # tibble (data frame) for testing
```

Model Induction and Testing

Use training set to build model, then predict insurance cost using the test set

```
model <- train(charges ~ age + bmi + smokerN + region_southeast,
               data = charges_train, # use training set
               method = "lm") # linear regression

# now predict outcomes in test set
p <- predict(model, charges_test)
```

Model Performance

Use training set to build model, then predict insurance cost using the test set

```
# how did we do? calculate performance across resamples
# RMSE and R-squared
postResample(pred = p, obs = charges_test$charges)
# on average, our prediction is off by $5,790.49
```

```
> postResample(pred = p, obs = charges_test$charges)
         RMSE     Rsquared          MAE
5790.4940335    0.7999239 4169.6005174
```

Model Performance

How to improve performance? One way is to try and specify a different method

```
model2 <- train(charges ~ age + bmi + smokerN + region_southeast,
               data = charges_train, # use training set
               method = "ranger") # random forest

# now predict outcomes in test set
p2 <- predict(model2, charges_test)

# how did we do? calculate performance across resamples
# RMSE and R-squared
postResample(pred = p2, obs = charges_test$charges)
# on average, our prediction is off by $4,093.32
```

```
> postResample(pred = p2, obs = charges_test$charges)
        RMSE     Rsquared          MAE
4093.3154611    0.8999708 2390.3930521
```

Which Model?

So many choices!

Linear Regression

```
method = 'lm'
```

Type: Regression

Tuning parameters:

- `intercept` (intercept)

A model-specific variable importance metric is available.

Random Forest

```
method = 'ranger'
```

Type: Classification, Regression

Tuning parameters:

- `mtry` (#Randomly Selected Predictors)
- `splitrule` (Splitting Rule)
- `min.node.size` (Minimal Node Size)

Required packages: `e1071`, `ranger`, `dplyr`

A model-specific variable importance metric is available.

http://topepo.github.io/caret/train-models-by-tag.html

7 `train` Models By Tag

The following is a basic list of model types or relevant characteristics. There entires in these lists are arguable. For example: random forests theoretically use feature selection but effectively may not, support vector machines use L2 regularization etc.

Contents

Which Model?

6 Available Models

The models below are available in `train`. The code behind these protocols can be obtained using the function `getModelInfo` or by going to the github repository.

Show 238 entries

Search:

Model	*method* Value	Type	Libraries	Tuning Paramete
Adaptive-Network-Based Fuzzy Inference System	ANFIS	Regression	frbs	num.labels, max.i
Bayesian Regularized Neural Networks	brnn	Regression	brnn	neurons
Bayesian Ridge Regression	bridge	Regression	monomvn	None
Bayesian Ridge Regression (Model Averaged)	blassoAveraged	Regression	monomvn	None
Cubist	cubist	Regression	Cubist	committees, neighbors

http://topepo.github.io/caret/available-models.html

Summary

- Regression with ML is different than regression with traditional OLS – one is focused on predictions while the other is focused on explanations
- When building a predictive ML model, split data into training and test sets (70-30 or 80-20)
- Always evaluate the performance of a model with the test data, and experiment with different methods to compare the performances of different models

Summary

- We use regression instead of correlation when we want to generate an equation that allows us to predict one variable from another (or from a set of) variable(s).

For every unit increase in X, how many units increase in Y can we expect?

- The "regression equation" is the equation that defines the "line of best fit"