BUDT 730
Data, Models and Decisions

Lecture 15

Regression Analysis (7)

Variable Selection

Prof. Sujin Kim

Practice: Prediction Model

Housing prices in MidCity

Data: HousePrices.xlsx

Example: Housing prices in MidCity

HousePrices.xlsx has data on 128 recent sales of single-family houses in MidCity.

Variables:

Price: Price at which house was eventually sold

SqFt: Floor area in square feet

Bedrooms: # of bedrooms

Bathrooms: # bathrooms

Offers: # offers made on the house prior to the accepted offer

Brick: Whether construction is primarily brick (yes/no)

Neighborhood: One of the three neighborhoods in MidCity (east, west or north)

Sample of data

Home	Price	SqFt	Bedrooms	Bathrooms	Offers	Brick	Neighborhood
1	114300	1790	2	2	2	No	East
2	114200	2030	4	2	3	No	East
3	114800	1740	3	2	1	No	East
4	94700	1980	3	2	3	No	East
5	119800	2130	3	3	3	No	East
6	114600	1780	3	2	2	No	North
7	151600	1830	3	3	3	Yes	West
8	150700	2160	4	2	2	No	West
9	119200	2110	4	2	3	No	East
10	104000	1730	3	3	3	No	East
11	132500	2030	3	2	3	Yes	East
12	123000	1870	2	2	2	Yes	East
13	102600	1910	3	2	4	No	North
14	126300	2150	3	3	5	Yes	North
15	176800	2590	4	3	4	No	West
16	145800	1780	4	2	1	No	West
17	147100	2190	3	3	4	Yes	East
18	83600	1990	3	3	4	No	North
19	111400	1700	2	2	1	Yes	East
20	167200	1920	3	3	2	Yes	West

Objective & data

- Objective: To predict the price of houses in MidCity.

- Data characterization
  - Y? X's?
  - Data size, dimension
  - Types of variables
  - Sample/Population?

Work stages

1. Understand the data (plots, descriptive statistics)
2. Partition the data:
   1. 70% training (can be 60-80%)
   2. 30% validation
3. Fit model(s) to training
4. Evaluate model(s) on test (validation)
5. Report conclusion

Data Partition

R functions:

- sort(): Sorting or Ordering Vectors, ex: sort(x, decreasing = FALSE, ...)
- sample(): Random Samples and Permutations: ex: sample(x, size)
- nrow(): The Number of Rows/Columns of an Array

```
# Splitting data
dt = sort(sample(nrow(HousePrices), nrow(HousePrices)*.7))
train<-HousePrices[dt,]
test<-HousePrices[-dt,]
```

Fit data to training

```
# Build a linear regression model
# Use all variables
Model1<-lm(Price~.,data=train)
summary(Model1)
observed<-test$Price
predicted<-predict(Model1,test)
```

Prediction

```
#Loading required R package
# Metrics, for mae,remes and mape
install.packages("Metrics")
library(Metrics)

# Compute MAE, RMSE, and MAPE
mae.Model1<-mae(observed,predicted)
rmse.Model1<-rmse(observed,predicted)
mape.Model1<-mape(observed,predicted)*100
print(c(mae.Model1,rmse.Model1,mape.Model1))
```

Model1: Price = ∑all

```
Call:
lm(formula = Price ~ ., data = train)
```

```
Residual standard error: 11220 on 81 degrees of freedom
Multiple R-squared:  0.8467,	Adjusted R-squared:  0.8335

F-statistic: 63.93 on 7 and 81 DF,  p-value: < 2.2e-16
```

```
> print(c(mae.Model1,rmse.Model1,mape.Model1))
[1] 5997.622878 7512.224838    4.735837
```

Model2: Price = ∑(all\offer)

```
Call:
lm(formula = Price ~ ., data = train[, -5])
```

```
Residual standard error: 13430 on 82 degrees of freedom
Multiple R-squared:  0.7775,	Adjusted R-squared:  0.7613
```

```
> print(c(mae.Model2,rmse.Model2,mape.Model2))
[1] 6991.277377 9612.513675    5.569141
```

Practice

- Build a model without offer.
- Introduce interactions.
- Build an exponential model.(log(Price))
- Identify the best.

Variable Selection

Selecting a Final Model

- As much of an *art* as it is a *science*
  - Gets better with experience!
- In practice, the choice of relevant independent variables is not obvious. Three guiding principles:
  - Domain knowledge or knowledge of theory
  - Data availability
    - Principle of **parsimony**: Explain the most with the least
  - Statistical inference
- Other considerations
  - Validation: How accurate is the model on data not used to fit the model?
  - What is the effect of outliers on our model?

Include/Exclude Decisions

- **Model selection** consists of a series of (independent) variable selection steps
- General guidelines
  - Use domain knowledge
  - Consider significance of the regression coefficients
    - Variables with p-values > 0.05 are candidates for exclusion.
  - Consider multicollinearity
  - Consider including/excluding several related variables as a group (common with categorical variables)

Automated Feature Selection

- There are automated methods to select features for regression model – Stepwise regression
  - Backward Elimination
  - Forward Selection
  - Stepwise: Forward + backward
- We specify “entry” and “exit” thresholds for a predictor to be added or removed from a model
  - Based on **p-values** or F-values

Stepwise Regression

Backward Elimination

- **Start with full model** using all independent variables
- Select least significant independent variable to remove
- Continue until no independent variables meet **removal criteria**

Forward Selection

- **Start with null model with only a constant**
- Select independent variable that adds the most explanatory value to the model
- Continue until no independent variables meet **selection criteria**

Stepwise

- **Start with a model with a base model**
- This is much like a forward procedure, except that it also considers possible deletions along the way
- Select independent variable that adds the most value to the model
- Search for any independent variable that meets the removal criteria
- Continue until no independent variables meet selection or removal criteria

Summary - Regression Modeling Process

- What are we trying to predict or understand?
  - What is the dependent variable?
- Explore the data!
  - Do we have the right data? What is the right set of independent variables?
  - Is the relationship linear? Apply transformations if necessary.
  - Are there potential interactions?
  - Are there outliers?
- Formulate and understand the model
  - Understand the economic interpretation of each coefficient, if possible
- Estimate the regression model
  - Are there variables that are not significant?
  - How can we improve the model?
  - Does the model meet our needs?
- Use the estimated regression model
  - Prediction, economical interpretation, decision-making support

Practice

Explanatory Model for Base Ball Data

`BaseBall Data.xlsx`

Base Ball Data

- Goal: Find the best explanatory regression model for “Salary”.

- Explore Salary data.
- Consider transformations of variables: dummy variables, nonlinear transformations, interactions
- Apply a stepwise regression (first try backward elimination)

Information Criteria: AIC and BIC

- AIC and BIC are used for comparing models
- Akaike Information Criterion (AIC):

$$AIC = 2k - 2Log(L^*)$$

  - k = number of parameters
  - L^*= maximum value of the likelihood function
  - In R, AIC for linear regression model is $n \log\left(\frac{RSS}{n}\right) + n + nlog(2\pi)$ and extracted AIC is $n \log\left(\frac{RSS}{n}\right)$. Step() uses extracted AIC, which is equivalent to use about 0.15 as a p-value.
- Bayesian Information Criterion (BIC):

$$BIC = \ln(n)k - 2Log(L^*)$$

  - n = sample size
- Given a set of candidate models for the data, the preferred model is the one with the **minimum AIC/BIC value.**

R packages and functions

- Packages:
  - car: for computing vif
  - MASS: for computing stepwise regression
- Functions
  - round: rounding of numbers
  - corrplot: for correlation plot
  - sapply(): apply a function over a vector or a list
  - is.numeric: test if an object is interpretable as numbers.
  - step(): stepwise variable selection. direction = “backward”, “forward”, “both”
  - AIC() or extractAIC(), , BIC()

Loading required R packages

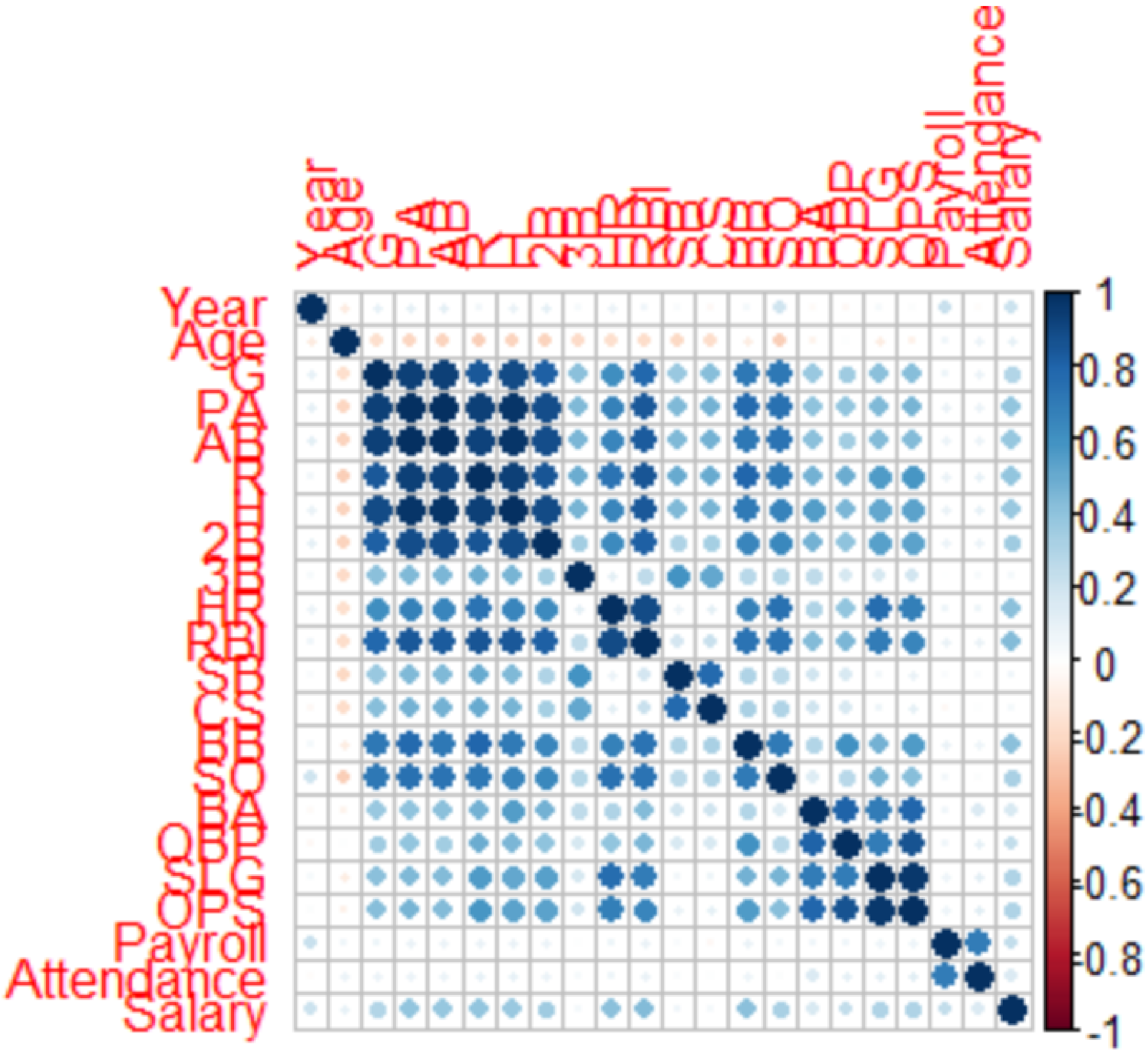
```
# corrplot, for correlation plot
install.packages("corrplot")
library(corrplot)

#car, for computing vif
install.packages("car")
library(car)

#MASS, for computing stepwise regression
install.packages("MASS")
library(MASS)
```

```r
#Remove Player and Team
df<-Baseball_Data[,-c(1,3)]
```

```r
# ModelAll
ModelAll<-lm(Salary~., data=df)
summary(ModelAll)
vif(ModelAll)
#Error in vif.default(model) : there are aliased coefficients in the model
#This error typically occurs when multicollinearity exists in a regression
```

```r
# select numeric variables & calculate the correlations
r <- cor(df[sapply(df,is.numeric)])
# rounded to 2 decimals
round(r,2)
#create a correlation plot
corrplot(r)
```

Correlation plot



Stepwise algorithms

```
#Backward elimination
StepBW<-step(ModelAll,direction = "backward")
summary(StepBW)
adjr2.StepBW<-summary(StepBW)$adj.r.squared
vif(StepBW)
```

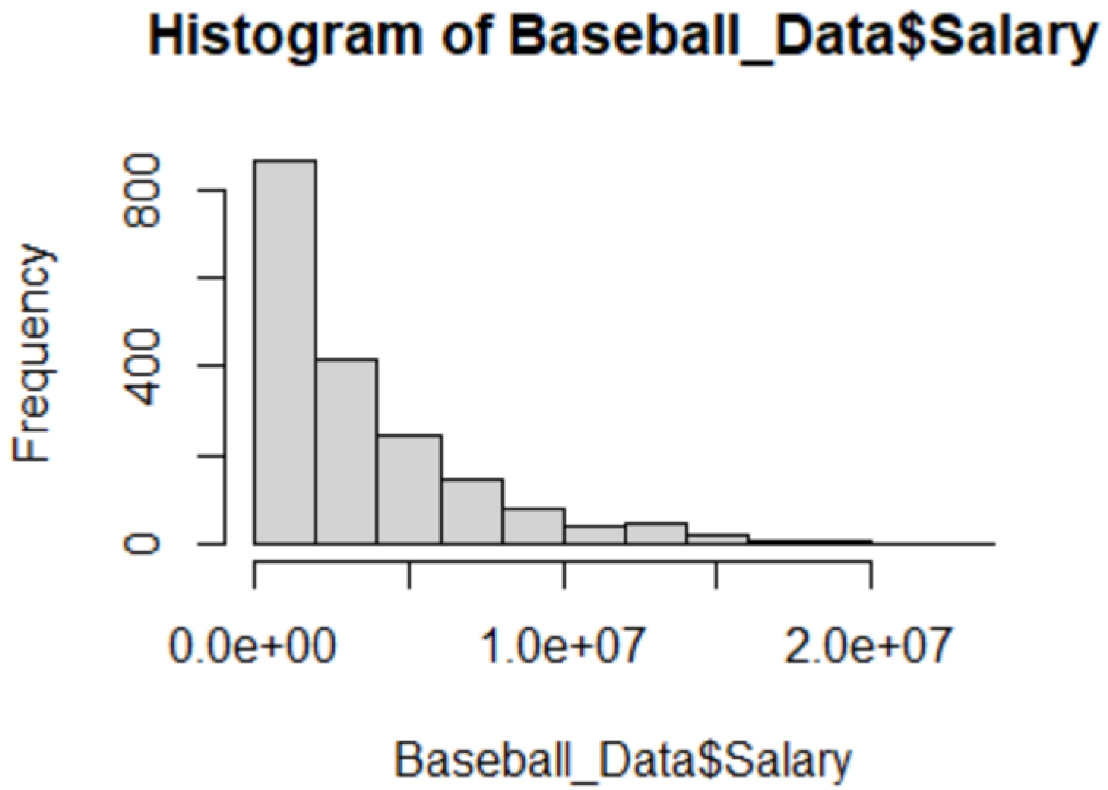
Create forward and stepwise model using direction = “forward” and “both”

```
AIC(StepBW, StepFW, StepW)
#or extractAIC(StepBW, StepFW, StepW)
BIC(StepBW, StepFW, StepW)
print(c(adjr2.StepBW, adjr2.StepFW, adjr2.StepW))
```

Comparisons

```
> AIC(StepBW, StepFW, StepW)
       df      AIC
StepBW 29 61192.14
StepFW 35 61200.10
StepW  29 61192.14
> #or extractAIC(StepBW, StepFW, StepW)
> BIC(StepBW, StepFW, StepW)
       df      BIC
StepBW 29 61352.75
StepFW 35 61393.93
StepW  29 61352.75
> print(c(adjr2.StepBW, adjr2.StepFW, adjr2.StepW))
[1] 0.4033604 0.4027067 0.4033604
```

Exponential models



Comparisons

```
> AIC(LogStepBW, LogStepFW, LogStepW)
          df      AIC
LogStepBW 29 4566.900
LogStepFW 35 4573.523
LogStepW  29 4566.900
> BIC(LogStepBW, LogStepFW, LogStepW)
          df      BIC
LogStepBW 29 4727.501
LogStepFW 35 4767.352
LogStepW  29 4727.501
> print(c(adjr2.LogStepBW, adjr2.LogStepFW, adjr2.LogStepW))
[1] 0.4367347 0.4365177 0.4367347
```

Next Time…

- Time Series Forecasting Models