Regression Analysis

Model Selection

Nicoleta Serban, Ph.D.

Professor

Stewart School of Industrial and Systems Engineering

Introduction

About This Lesson

Objectives

- **High Dimensionality:** When we have a very large number of predicting variables to consider, it can be difficult to interpret and work with the fitted model.

- **Multicollinearity:** When the predicting variables are correlated, it is important to select variables in such a way that the impact of multicollinearity is minimized.

- **Prediction vs Explanatory Objective:** The variables selected for the two objectives will most often be different.

➔ **Variable Selection** addresses all these concerns.

Georgia Tech

Implications and Words of Caution

- **Controlling vs. Explanatory Variables**
  - Consider research hypothesis as well as potential controlling variables

- **Targeted Predicting Variables**
  - Include target variable in model if specified by research hypothesis

- **Over-Interpretation**
  - Selected variables are not necessarily special!
    - Highly influenced by correlations between variables
    - Interpretation of regression coefficients
    - Causality vs. Association

No Magic Bullet

- Variable selection for large number of predicting variables is an “unsolved” problem in statistics
- In some sense, model selection is “data mining”
- Data miners / machine learners often work with many predictors
- There are no magic procedures to get you the “best model”

“All models are wrong, but some are useful.” —George Box

Georgia Tech

Notation

Given

$S \subset \{1, \ldots, p\}$ a subset of indices

and

(x_j for $j \in S$) the subset of predicting variables with indices in S:

- $\widehat{\boldsymbol{\beta}}(S)$ is the vector of estimated regression coefficients for the submodel with $\boldsymbol{X}_S = (x_j$ for $j \in S)$ predicting variables

- $\widehat{\boldsymbol{Y}}(S)$ is the vector of fitted values for the submodel with $\boldsymbol{X}_S = (x_j$ for $j \in S)$ predicting variables
  - E.g., for regression assuming normality, $\widehat{\boldsymbol{Y}}(S) = \boldsymbol{X}_S\widehat{\boldsymbol{\beta}}(S)$

➔ I will refer to this model as the **S submodel**.

Summary

Regression Analysis

Model Selection

Nicoleta Serban, Ph.D.

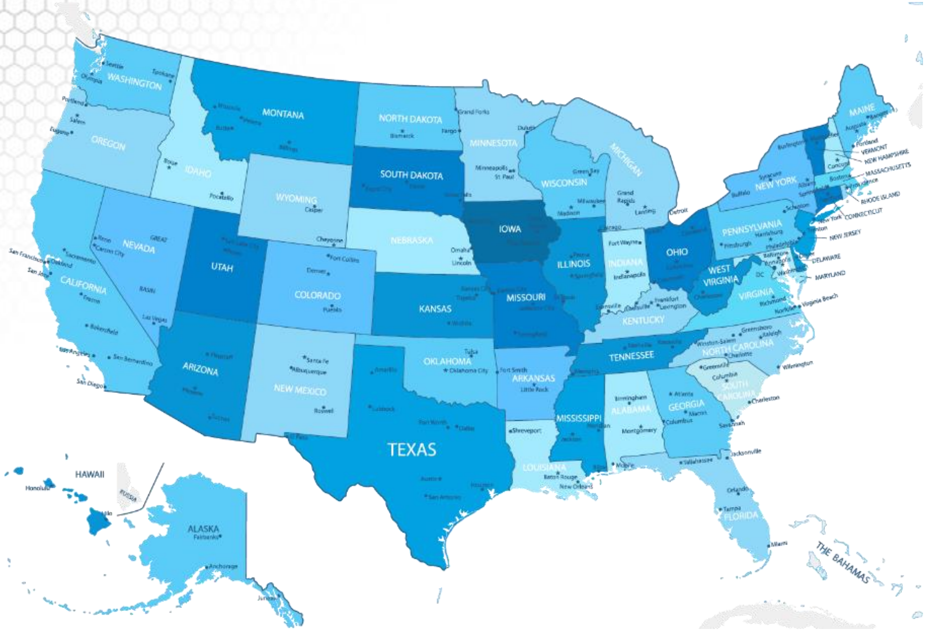
Professor

Stewart School of Industrial and Systems Engineering

Data Examples

About This Lesson

Ranking States by SAT Performance



SAT Mean Score by State – Year 1982
790 (South Carolina) – 1088 (Iowa)

- ***Which variables are associated with state average SAT scores?***
- ***After accounting for selection biases, how do the states rank?***
- ***Which states perform best for the amount of money they spend?***

Response & Predicting Variables

The response variable is:

Y = State average SAT score (verbal and quantitative combined)

The predicting variables are:

takers	% of eligible students (high school seniors) in state who took the exam
rank	Median percentile ranking of test takers in their secondary school classes
income	Median income of families of test takers (in $00's)
years	Average years test takers had in social/natural sciences and humanities
public	% of test takers who attended public schools
expend	State expenditure on secondary schools (in $00's/student)

Regression Analysis

regression.line = lm(sat ~ log(takers) + rank + income + years + public + expend)
summary(regression.line)

Coefficients:

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	407.53990	282.76325	1.441	0.15675
log(takers)	-38.43758	15.95214	-2.410	0.02032 *
rank	4.11427	2.50166	1.645	0.10734
income	-0.03588	0.13011	-0.276	0.78407
years	17.21811	6.32007	2.724	0.00928 **
public	-0.11301	0.56239	-0.201	0.84168
expend	2.56691	0.80641	3.183	0.00271 **

Shall we discard the predicting variables with regression coefficients that are not statistically significant?

➔ NO. Perform variable selection.

Test for Statistical Significance

p-values

$\hat{\beta}_{takers} \approx 0.02$

$\hat{\beta}_{rank} > 0.1$

$\hat{\beta}_{income} > 0.1$

$\hat{\beta}_{years} < 0.01$

$\hat{\beta}_{public} > 0.1$

$\hat{\beta}_{expend} < 0.01$

Inference on Subset of Coefficients

regression.red = lm(sat ~ log(takers) + rank)
anova(regression.red, regression.line)

Model 1: sat ~ log(takers) + rank
Model 2: sat ~ log(takers) + rank + income + years + public + expend

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	47	45530				
2	43	26585	4	18945	7.6604	9.42e-05 ***

Testing for a subset of regression coefficients:

H_0: Reduced Model (takers and rank only)

vs.

H_A: Full Model

Partial F Test: F-value = 7.6604, P-value ≈ 0

Inference on Subset of Coefficients

regression.red = lm(sat ~ log(takers) + rank)
anova(regression.red, regression.line)

Model 1: sat ~ log(takers) + rank
Model 2: sat ~ log(takers) + rank + income + years + public + expend

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	47	45530				
2	43	26585	4	18945	7.6604	9.42e-05 ***

- **Controlling and explanatory variables**: log(takers) and rank need to be in the model.
- **Partial F test for explanatory variables**: at least one predicting variable has explanatory power. Which ones?

➔ Perform variable selection!!!

Predicting Bankruptcy

- Effective bankruptcy prediction is useful for investors and analysts, allowing for accurate evaluation of a firm's prospects.

- Roughly 40 years ago, Ed Altman showed that publicly available financial indicators can be used to distinguish between firms that are about to go bankrupt and those that are not.

Which financial indicators are associated with bankruptcy for telecommunications firms?

Acknowledgement: This example was provided by Dr. Jeffrey Simonoff from New York University and was inspired by the honors thesis of Jeffrey Lui.

Georgia Tech

Bankruptcy Data

Data Sample:

- 25 telecommunication firms that declared bankruptcy 2000–2002
- 25 telecommunication firms that did not declare bankruptcy, "matched" according to the asset size of the bankrupt firms

Replicate Experimental Data Setting:

→ Matching firms to be comparable with respect to meaningful factors

→ Allowing for causal inference

Response & Predicting Variables

The response variable is:

Y = Whether the firm declared bankruptcy

The predicting variables are:

WC.TA Working capital as a percentage of total assets (in %)

RE.TA Retained earnings as a percentage of total assets (in %)

EBIT.TA Earnings before interest and taxes as a percentage of total assets (in %)

S.TA Sales as a percentage of total assets (in %)

BVE.BVL Book value of equity divided by book value of total liabilities

Exploratory Data Analysis

Read the data from the file## Exploratory analysis

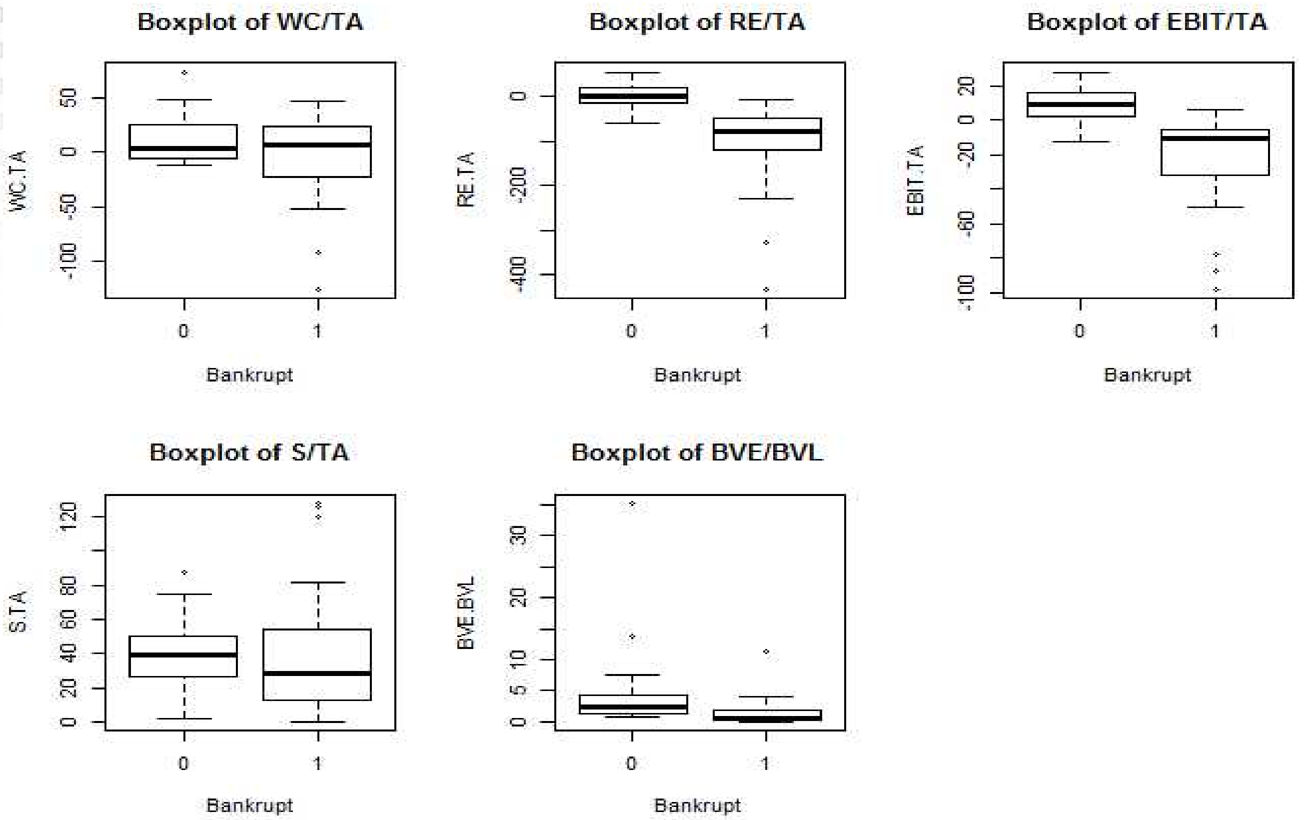
```
bankruptcy = read.table("bankruptcy.dat", sep="\t", header=T, row.names=NULL)
attach(bankruptcy)
```

Exploratory analysis

```
par(mfrow=c(2,3))

boxplot(split(WC.TA,Bankrupt), style.bxp="old", xlab="Bankrupt", ylab="WC.TA",
main="Boxplot of WC/TA")

boxplot(split(RE.TA,Bankrupt), style.bxp="old", xlab="Bankrupt", ylab="RE.TA",
main="Boxplot of RE/TA")

boxplot(split(EBIT.TA,Bankrupt), style.bxp="old", xlab="Bankrupt", ylab="EBIT.TA",
main="Boxplot of EBIT/TA")

boxplot(split(S.TA,Bankrupt), style.bxp="old", xlab="Bankrupt", ylab="S.TA",
main="Boxplot of S/TA")

boxplot(split(BVE.BVL,Bankrupt), style.bxp="old", xlab="Bankrupt", ylab="BVE.BVL",
main="Boxplot of BVE/BVL")
```

Georgia Tech

Exploratory Data Analysis



Regression Analysis

bank1 = glm(Bankrupt ~ WC.TA + RE.TA + EBIT.TA + S.TA + BVE.BVL, family=binomial)
summary(bank1)

Coefficients:

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	7.42646	6.35770	1.168	0.243
WC.TA	-0.15587	0.12208	-1.277	0.202
RE.TA	-0.07605	0.06311	-1.205	0.228
EBIT.TA	-0.49111	0.32260	-1.522	0.128
S.TA	-0.08040	0.09216	-0.872	0.383
BVE.BVL	-2.07764	1.47488	-1.409	0.159

gstat = bank1$null.deviance - deviance(bank1)
cbind(gstat, 1 - pchisq(gstat, length(coef(bank1))-1))

```
        gstat
[1,] 57.46799 4.049594e-11
```

Test for Statistical Significance

All p-values > 0.1

None of the coefficients are statistically significant.

Test for Overall Regression

p-value ≈ 0

The overall regression has predictive power.

Georgia Tech

Summary

Regression Analysis
Model Selection

Nicoleta Serban, Ph.D.
Professor
Stewart School of Industrial and Systems Engineering

Prediction Risk Estimation

About This Lesson

Bias-Variance Tradeoff

- **Variable Selection:** Bias vs. Variance
  - Many covariates
    - Low bias, high variance
  - Few covariates
    - High bias, low variance

- **Prediction Risk:** Measure of the Bias-Variance Tradeoff

$$R(S) = \frac{1}{n}\sum_{i=1}^{n} E(\widehat{\boldsymbol{Y}}_i(S) - \boldsymbol{Y}_i^*)^2$$

with $\widehat{\boldsymbol{Y}}_i(S)$ the fitted response for submodel S and $\boldsymbol{Y}_i^*$ the future observation

We cannot obtain the prediction risk because we do not have the future observations.

How to estimate?

Training Risk

- Replace future observations with actual observations

$$R_{\mathrm{tr}}(\mathrm{S}) = \frac{1}{n}\sum_{i=1}^{n}(\widehat{\boldsymbol{Y}}_i(\mathrm{S}) - \boldsymbol{Y}_i)^2$$

with $\widehat{\boldsymbol{Y}}_i(S)$ the fitted response for submodel S and $\boldsymbol{Y}_i$ the actual observation

- Uses data twice (data snooping): downward bias in prediction risk estimate

- Always prefers/selects larger/more complex model

➔ **Correcting for the bias**

$$R_{\mathrm{tr}}(\mathrm{S}) + Complexity\ Penalty$$

Variable Selection Criteria

➔ **Correcting for the bias:** $R_{\text{tr}}(S) + Complexity\ Penalty$

➔ **Selection criteria** differ through the complexity penalty as follows**:**

- **Mallow's Cp** with $Complexity\ Penalty = \dfrac{2|S|\hat{\sigma}^2}{n}$

  where |S| is the model size (number of predictors) and $\hat{\sigma}^2$ is the estimated variance based on the full model.

- **Akaike Information Criterion (AIC)** with $Complexity\ Penalty = \dfrac{2|S|\sigma^2}{n}$
  where |S| is the model size and σ^2 is the true variance.

> - For AIC, we need to replace σ^2 with an estimate (from the full model or from the S submodel).

Variable Selection Criteria (cont'd)

➔ **Correcting for the bias:** $R_{\mathrm{tr}}(S) + Complexity\ Penalty$

➔ **Selection criteria** differ through the complexity penalty as follows**:**

- **Bayesian Information Criterion (BIC)** with

$$Complexity\ Penalty = \frac{|\mathrm{S}|\sigma^2 \log(n)}{n}$$

where |S| is the model size and σ^2 is the true variance

- For BIC, we need to replace σ^2 with an estimate (from the full model or from the S submodel)
- BIC penalizes complexity more than other approaches
  - Preferred in model selection for prediction

Variable Selection Criteria (cont'd)

➔ **Correcting for the bias:** $R_{\text{tr}}(S) + Complexity\ Penalty$

- **Leave-one-out Cross Validation**

$$R_{\text{CV}}(S) = \frac{1}{n}\sum_{i=1}^{n}\left(\widehat{\boldsymbol{Y}}_{(i)}(\text{S}) - \boldsymbol{Y}_i\right)^2$$

where $\widehat{\boldsymbol{Y}}_{(i)}(S)$ is the i-th predicted value from the S submodel without the i-th observation

- **Leave-one-out Cross Validation Approximation**

$$\hat{R}_{\text{CV}}(\text{S}) \approx R_{\text{tr}}(\text{S}) + \frac{2|\text{S}|\hat{\sigma}^2(\text{S})}{n}$$

where $\hat{\sigma}^2(S)$ is the estimated variance based on the S submodel.

Generalized Linear Models

Training Risk for Generalized Linear Models (including for logistic regression and Poisson regression)

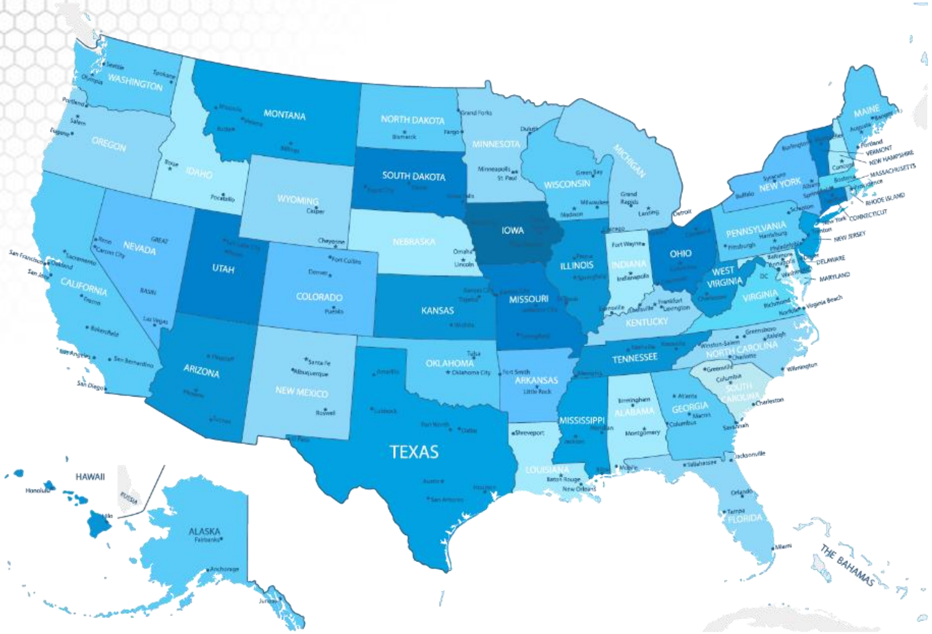
$$R_{\mathrm{tr}}(\mathrm{S}) = \frac{1}{n}\sum_{i=1}^{n}(2\boldsymbol{Y}_i\log[\boldsymbol{Y}_i/\widehat{\boldsymbol{Y}}_i(S)]+2(n_i-\boldsymbol{Y}_i)\log[(n_i-\boldsymbol{Y}_i)/(n_i-\widehat{\boldsymbol{Y}}_i(\mathrm{S}))])$$

where $\widehat{\boldsymbol{Y}}_i(S)$ the fitted response for submodel S and $\boldsymbol{Y}_i$ the actual observation

➔ **Correcting for the bias:** $R_{\mathrm{tr}}(\mathrm{S}) + Complexity\ Penalty$

- AIC & BIC are commonly used for model selection for GLMs

Ranking States by SAT Performance



SAT Mean Score by State – Year 1982
790 (South Carolina) – 1088 (Iowa)

- ***Which variables are associated with state average SAT scores?***
- ***After accounting for selection biases, how do the states rank?***
- ***Which states perform best for the amount of money they spend?***

Georgia Tech

Model Selection Criteria Using R

```
library(CombMSC)
n = nrow(datasat)

## full model
c(Cp(regression.line, S2=summary(regression.line)$sigma^2),
AIC(regression.line, k=2), AIC(regression.line,k=log(n)))
[1]   7.016756 471.698197 486.994381

## reduced model
c(Cp(regression.red, S2=summary(regression.line)$sigma^2),
AIC(regression.red, k=2), AIC(regression.red,k=log(n)))
[1]  29.67045 490.59880 498.24689
```

- Mallow's Cp: $\hat{\sigma}$ = 24.86 is the estimated standard deviation for the full model
  - Use the estimated variance, $\hat{\sigma}$^2, as the S2 parameter value
- BIC: Similar to AIC, but the AIC complexity is further penalized by log(n)
- The values of the three criteria are different and not comparable
- The full model is better according to all three criteria

Summary

Regression Analysis

Model Selection

Nicoleta Serban, Ph.D.

Professor

Stewart School of Industrial and Systems Engineering

Model Search

About This Lesson

Bias-Variance Tradeoff

- **Variable Selection:** Bias vs. Variance
  - Many covariates
    - Low bias, high variance
  - Few covariates
    - High bias, low variance
  - Too few covariates
    - High bias, high variance
- **Prediction Risk:** Measure of the Bias-Variance Tradeoff

$$R(S) = \frac{1}{n}\sum_{i=1}^{n} E(\widehat{\boldsymbol{Y}}_i(S) - \boldsymbol{Y}_i^*)^2$$

with $\widehat{\boldsymbol{Y}}_i(S)$ the fitted response for submodel S and $\boldsymbol{Y}_i^*$ the future observation

Given an estimate of the prediction risk for a submodel S, choose the submodel with the smallest prediction risk.

➔ ***How to search over all submodels?***

Model Search

- If p is the number of predicting variables, there are $\mathbf{2^p}$ possible submodels
  - If p is small
    - Fit all submodels
  - If p is large
    - Search using heuristics/greedy search

- **Stepwise Regression**
  - Forward
    - Start with no predictors, add one at a time
  - Backward
    - Start with all predictors, drop one at a time
  - Forward-Backward
    - Add and drop one variable at a time iteratively

Model Search

- Stepwise regression is a greedy algorithm. It does not guarantee to find the model with the best score.
- Forward stepwise regression is preferable to backward stepwise regression.
- Forward stepwise regression does not necessarily select the same model as the one selected using backward stepwise regression.

Forward Stepwise Regression

1. Select criterion for model selection (e.g., AIC)
2. Establish minimum model, and compute its criterion value, C_0
3. Fit p marginal regressions for p predictors, V_j $(j = 1, \cdots, p)$, that are not in minimum model
   - C_j is the criterion value for the model that includes the j-th predictor, V_j
   - If possible, select predictor $P_1 = V_k$ whose inclusion yields the <u>smallest</u> criterion value where $C_k < C_0$
   - If P_1 exists, add it to the minimum model and continue; otherwise, stop
4. Fit p-1 regressions, and use the same method to test if another predictor should be added
   - Regressions will now be based on models with the previous predictors, including P_1, and with each V_j additionally included one at a time, for $j = 1, \cdots, (k-1), (k+1), \cdots, p$
   - If possible, select predictor $P_2 = V_l$ whose inclusion yields the <u>smallest</u> criterion value where $C_l < C_k$
     - C_l is based on the current regressions; C_k is based on the regressions from the previous step
   - If P_2 exists, add it to the model and continue; otherwise, stop
5. Continue adding predictors one at a time until the criterion does not improve

Backward Stepwise Regression

1. Select criterion for model selection (e.g., AIC)
2. Establish the minimum model and the predictors that must be included
3. Fit full model with p additional predictors not in the minimum model, V_j $(j = 1, \cdots, p)$, and compute its criterion value, C_{F}
4. Fit p regressions, removing one predictor, V_j $(j = 1, \cdots, p)$, each time
   - C_j is the criterion value for the model that excludes the j-th predictor, V_j
   - If possible, select predictor $P_1 = V_k$ whose removal yields the <u>smallest</u> criterion value where $C_k \le C_{\mathrm{F}}$
   - If P_1 exists, remove it from the full model and continue; otherwise, stop
5. Fit p-1 regressions, and use the same method to test if another predictor should be removed
   - Regressions will now be based on models with the previous predictors, excluding P_1, and with each remaining V_j removed one at a time, for $j = 1, \cdots, (k-1), (k+1), \cdots, p$
   - If possible, select $P_2 = V_l$ whose removal yields the <u>smallest</u> criterion value where $C_l \le C_k$
     - C_l is based on the current regressions; C_k is based on the regressions from the previous step
   - If P_2 exists, remove it from the model and continue; otherwise, stop
6. Continue discarding predictors one at a time until the criterion does not improve

Georgia Tech

Forward vs Backward Stepwise Regression

Backward stepwise regression:

- Cannot be performed if there are more predictors than the sample size $(p > n)$
- Is more computationally expensive than forward stepwise regression
- Will select larger models if p is large

Summary

Regression Analysis
Model Selection

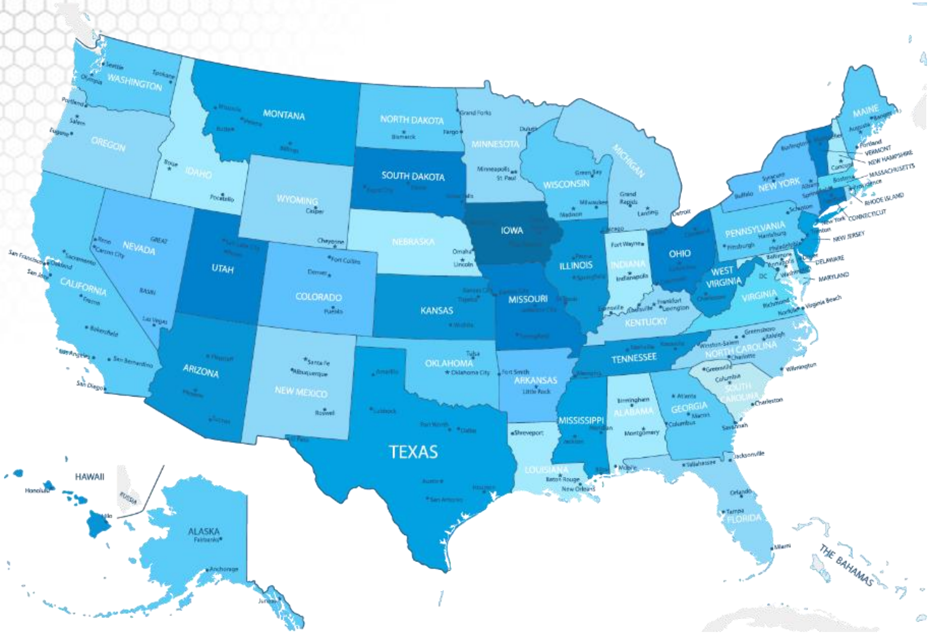
Nicoleta Serban, Ph.D.
Professor
Stewart School of Industrial and Systems Engineering

Model Search: Data Examples

Georgia Tech

About This Lesson

Ranking States by SAT Performance



SAT Mean Score by State – Year 1982
790 (South Carolina) – 1088 (Iowa)

- ***Which variables are associated with state average SAT scores?***
- ***After accounting for selection biases, how do the states rank?***
- ***Which states perform best for the amount of money they spend?***

Compare All Models

```
library(leaps)
out = leaps(datasat[,-c(1,2)], sat, method = "Cp")
cbind(as.matrix(out$which),out$Cp)
  1 2 3 4 5 6
1 0 0 0 0 0 1   34.026834
1 1 0 0 0 0 0   47.639512
1 0 1 0 0 0 0 187.387572
1 0 0 1 0 0 0 269.647903
1 0 0 0 1 0 0 306.188562
1 0 0 0 0 1 0 307.076043
⋮
6 1 1 1 1 1 1   7.000000

best.model = which(out$Cp==min(out$Cp))
cbind(as.matrix(out$which), out$Cp)[best.model,]
       1        2        3        4        5        6
0.000000 0.000000 1.000000 1.000000 1.000000 1.000000 3.581157
```

The output includes all 64 combinations of predictors with specification of which predictors are in the model and the Cp score value for each model.

The best model with respect to Mallow’s Cp criterion:
years, public, expend, rank (last four predictors in the input dataset)

Does not allow for specification of controlling variables!!!

Stepwise Regression

 # Forward Stepwise Regression

step(lm(sat~log(takers)+rank), scope=list(lower=sat~log(takers)+rank,
upper=sat~log(takers)+rank+expend+years+income+public), direction="forward")

Start: AIC=346.7
sat ~ log(takers) + rank

	Df	Sum of Sq	RSS	AIC
+ expend	1	13149.5	32380	331.66
+ years	1	9827.2	35703	336.55
<none>			45530	346.70
+ income	1	1305.3	44224	347.25
+ public	1	15.9	45514	348.69

Step: AIC=331.66
sat ~ log(takers) + rank + expend

	Df	Sum of Sq	RSS	AIC
+ years	1	5743.5	26637	323.90
<none>			32380	331.66
+ public	1	421.0	31959	333.01
+ income	1	317.3	32063	333.17

Step: AIC=323.9
sat ~ log(takers) + rank + expend + years

	Df	Sum of Sq	RSS	AIC
<none>			26637	323.90
+ income	1	26.6165	26610	325.85
+ public	1	4.5743	26632	325.89

Call:
lm(formula = sat ~ log(takers) + rank + expend + years)

Coefficients:

(Intercept)	log(takers)	rank	expend	years
388.425	-38.015	4.004	2.423	17.857

Selected model: *expend* and *years*, with confounding variables log(*takers*) and *rank*

Stepwise Regression (cont'd)

 # Backward Stepwise Regression

full = lm(sat ~ log(takers) + rank + expend + years + income + public)
minimum = lm(sat ~ log(takers) + rank)
step(full, scope=list(lower=minimum, upper=full), direction="backward")

Start: AIC=327.8
sat ~ log(takers) + rank + expend + years + income + public

	Df	Sum of Sq	RSS	AIC
- public	1	25.0	26610	325.85
- income	1	47.0	26632	325.89
<none>			26585	327.80
- years	1	4588.8	31174	333.77
- expend	1	6264.4	32850	336.38

Step: AIC=325.85
sat ~ log(takers) + rank + expend + years + income

	Df	Sum of Sq	RSS	AIC
- income	1	26.6	26637	323.90
<none>			26610	325.85
- years	1	5452.8	32063	333.17
- expend	1	7430.3	34040	336.16

Step: AIC=323.9
sat ~ log(takers) + rank + expend + years

	Df	Sum of Sq	RSS	AIC
<none>			26637	323.90
- years	1	5743.5	32380	331.66
- expend	1	9065.8	35703	336.55

Call:
lm(formula = sat ~ log(takers) + rank + expend + years)

Coefficients:

(Intercept)	log(takers)	rank	expend	years
388.425	-38.015	4.004	2.423	17.857

Stepwise Regression (cont'd)

```
# Backward Stepwise Regression
full = lm(sat ~ log(takers) + rank + expend + years + income + public)
minimum = lm(sat ~ log(takers) + rank)
step(full, scope=list(lower=minimum, upper=full), direction="backward")
```

Start: AIC=327.8
sat ~ log(takers) + rank + expend + years + income + public

	Df	Sum of Sq	RSS	AIC
- public	1	25.0	26610	325.85
- income	1	47.0	26632	325.89
<none>			26585	327.80
- years	1	4588.8	31174	333.77
- expend	1	6264.4	32850	336.38

Step: AIC=325.85
sat ~ log(takers) + rank + expend + years + income

	Df	Sum of Sq	RSS	AIC
- income	1	26.6	26637	323.90
<none>			26610	325.85
- years	1	5452.8	32063	333.17
- expend	1	7430.3	34040	336.16

- Selected model includes
  - *expend* and *years*
  - confounding variables log(*takers*) and *rank*
- The same model was selected using forward regression
  - Generally, for a large number of predictors, the two methods will select different models

Step: AIC=323.9
sat ~ log(takers) + rank + expend + years

	Df	Sum of Sq	RSS	AIC
<none>			26637	323.90
- years	1	5743.5	32380	331.66
- expend	1	9065.8	35703	336.55

Call:
lm(formula = sat ~ log(takers) + rank + expend + years)

Coefficients:

(Intercept)	log(takers)	rank	expend	years
388.425	-38.015	4.004	2.423	17.857

Georgia Tech

Predicting Bankruptcy

- Effective bankruptcy prediction is useful for investors and analysts, allowing for accurate evaluation of a firm's prospects.

- Roughly 40 years ago, Ed Altman showed that publicly available financial indicators can be used to distinguish between firms that are about to go bankrupt and those that are not.

Which financial indicators are associated with bankruptcy for telecommunications firms?

Acknowledgement: This example was provided by Dr. Jeffrey Simonoff from New York University and was inspired by the honors thesis of Jeffrey Lui.

Compare All Models

```
library(bestglm)
input.Xy <- as.data.frame(cbind(WC.TA, RE.TA, EBIT.TA, S.TA,
BVE.BVL,Bankrupt))
bestBIC <- bestglm(input.Xy, IC="BIC", family=binomial)

bank2 = glm(Bankrupt~RE.TA+EBIT.TA+BVE.BVL,
family=binomial, epsilon=1e-14, maxit=500, x=T)
summary(bank2)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-0.29478	1.12323	-0.262	0.7930
RE.TA	-0.05627	0.02745	-2.050	0.0404 *
EBIT.TA	-0.16763	0.09270	-1.808	0.0706 .
BVE.BVL	-0.62975	0.39435	-1.597	0.1103

The best model selected with respect to BIC: *RE.TA, EBIT.TA, BVE.BVL*

- *RE.TA* is now statistically significant at $\alpha = 0.05$
- Not all coefficients are statistically significant

- RE.TA is associated with a decrease in the odds of going bankrupt in the next year by 5.5% holding all else fixed
- EBIT.TA is associated with a decrease in the odds of going bankrupt by 15%

Georgia Tech

Compare All Models (cont'd)

```
# Testing for subset of regression coefficients
gstat = deviance(bank2) - deviance(bank1)
cbind(gstat, 1-pchisq(gstat,length(coef(bank1))-length(coef(bank2))))
        gstat
[1,] 4.040336 0.1326332
```

The null (reduced model) is not rejected

Remove Outlier

bankrupt2 = bankruptcy[-1,]
attach(bankrupt2)
bank3 = glm(Bankrupt ~ WC.TA + RE.TA + EBIT.TA + S.TA + BVE.BVL, family=binomial,
maxit=500, data=bankrupt2)
Warning message:
glm.fit: fitted probabilities numerically 0 or 1 occurred

summary(bank3)
Coefficients:

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	265.467	576281.709	0	1
WC.TA	-4.297	12439.717	0	1
RE.TA	-1.516	5131.146	0	1
EBIT.TA	-17.043	35543.170	0	1
S.TA	-2.859	7408.747	0	1
BVE.BVL	-77.540	184903.001	0	1

The model fits perfectly. This is complete separation, and the solution is to simplify the model if that is possible.

Compare All Models: Without Outlier

input.Xy <- as.data.frame(cbind(WC.TA, RE.TA, EBIT.TA, S.TA, BVE.BVL,Bankrupt))
bestBIC <- bestglm(input.Xy, IC="BIC", family=binomial)

bank4 = glm(Bankrupt ~ RE.TA + EBIT.TA + BVE.BVL, family=binomial, maxit=500)
summary(bank4)
Coefficients:

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-0.09166	1.47135	-0.062	0.9503
RE.TA	-0.08229	0.04230	-1.945	0.0517 .
EBIT.TA	-0.26783	0.15854	-1.689	0.0912 .
BVE.BVL	-1.21810	0.76536	-1.592	0.1115

exp(coef(bank2)[-1])

RE.TA	EBIT.TA	BVE.BVL
0.9452862	0.8456655	0.5327273

exp(coef(bank4)[-1])

RE.TA	EBIT.TA	BVE.BVL
0.9210091	0.7650371	0.2957930

The best model selected with respect to BIC:
WC.TA, *RE.TA, EBIT.TA, BVE.BVL*

Stepwise Regression: Without Outlier

```
bank3.select=step(bank3, direction="backward")
summary(bank3.select)
```

Start: AIC=12
Bankrupt ~ WC.TA + RE.TA + EBIT.TA + S.TA + BVE.BVL

	Df	Deviance	AIC
- S.TA	1	0.0000	10.000
<none>		0.0000	12.000
- WC.TA	1	9.3839	19.384
- RE.TA	1	10.7362	20.736
- EBIT.TA	1	14.7992	24.799
- BVE.BVL	1	19.0267	29.027

Coefficients:

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	255.413	728539.823	0	1
WC.TA	-9.542	23920.936	0	1
RE.TA	-5.152	15669.825	0	1
EBIT.TA	-28.983	90578.211	0	1
BVE.BVL	-103.614	225264.760	0	1

Step: AIC=10
Bankrupt ~ WC.TA + RE.TA + EBIT.TA + BVE.BVL

	Df	Deviance	AIC
<none>		0.0000	10.000
- WC.TA	1	9.3841	17.384
- RE.TA	1	12.8531	20.853
- EBIT.TA	1	14.8672	22.867
- BVE.BVL	1	19.1321	27.132

Stepwise regression selects the same four predictors as the best subset selection approach using BIC.

Georgia Tech

Summary

Regression Analysis

Model Selection

Nicoleta Serban, Ph.D.

Professor

Stewart School of Industrial and Systems Engineering

Regularized Regression: Penalties

Georgia Tech

About This Lesson

Bias-Variance Tradeoff

Prediction Risk: Measure of the Bias-Variance Tradeoff

$$R(S) = \frac{1}{n}\sum_{i=1}^{n} E(\widehat{\boldsymbol{Y}}_i(S) - \boldsymbol{Y}_i^*)^2$$

Irreducible error

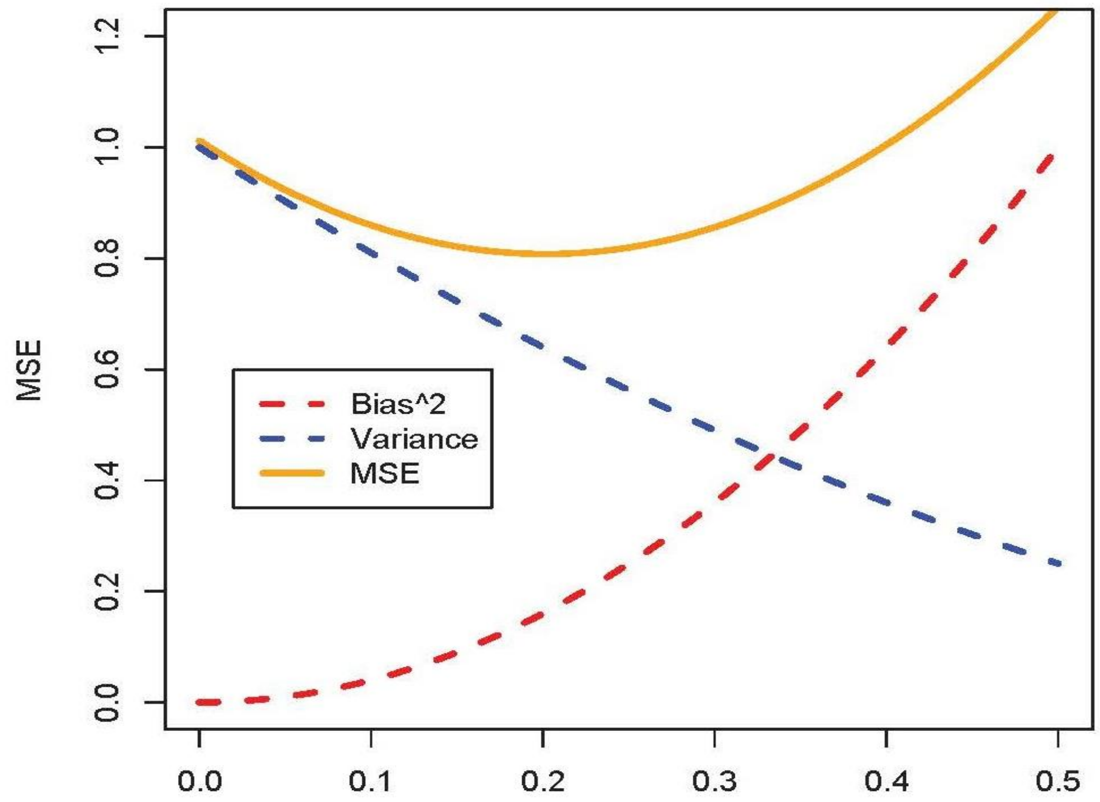
Mean Square Error

$$= V(\boldsymbol{Y}_i^*) + Bias^2\left(\widehat{\boldsymbol{Y}}_i(S)\right) + V(\widehat{\boldsymbol{Y}}_i(S))$$

for a submodel S, with $\widehat{\boldsymbol{Y}}_i(S)$ the fitted response for model S and $\boldsymbol{Y}_i^*$ the future observation.

- It is possible to find a model with lower MSE than the full model!
- It is “generic” in statistics: introducing some bias often yields in a decrease in MSE.

Bias-Variance Tradeoff



Biased Regression: Penalties

Not all biased models are better.

We need a way to find "good" biased models!

- Penalize large values of βs jointly
  - Should lead to "multivariate" shrinkage of the vector $\boldsymbol{\beta}$
- Goal is really to penalize "complex" models
  - Heuristically, "large" is interpreted as "complex model"
    - If truth really is complex, this may not work!
      - It will then be hard to build a good model anyways

Georgia Tech

Regularized Regression

Without Penalization

Estimate (β_0, β_1,…, β_p) by minimizing the sum of squared errors

$$\sum_{i=1}^{n}\left(y_i - (\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip})\right)^2$$

With Penalization

Estimate (β_0, β_1,…, β_p) by minimizing the penalized sum of squared errors

$$\sum_{i=1}^{n}\left(y_i - (\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip})\right)^2 + \lambda Penalty(\beta_1, \ldots, \beta_p)$$

The bigger lambda(λ), the bigger the penalty for model complexity.

Regularized Regression (cont'd)

The penalized sum of squared errors:

$$Q(\beta_1, \dots, \beta_p) = \sum_{i=1}^{n} \left(y_i - \left(\beta_0 + \beta_1 x_{i1} + \dots + \beta_p x_{ip} \right) \right)^2 + \lambda Penalty(\beta_1, \dots, \beta_p)$$

We consider three choices for the penalty:

$\boldsymbol{L_0}$ **penalty**

$||\beta||_0 = \#\{j\colon \beta_j \neq 0\}$ ⇒ Minimizing Q means searching through all submodels

$\boldsymbol{L_1}$ **penalty** (LASSO Regression)

$||\beta||_1 = \sum_{j=1}^{p} |\beta_j|$ ⇒ Minimizing Q forces many β_js to be zeros

$\boldsymbol{L_2}$ **penalty** (Ridge Regression)

$||\beta||_2 = \sum_{j=1}^{p} {\beta_j}^2$ ⇒ Minimizing Q accounts for multicollinearity

Comparing Penalties

- $\boldsymbol{L}_0$ penalty
  - Provides best model given a selection criterion
  - Requires fitting all submodels
- $\boldsymbol{L}_1$ penalty
  - Measures sparsity
- $\boldsymbol{L}_2$ penalty
  - Easy to implement
  - Does not do variable selection

Example: Consider vectors $\boldsymbol{u} = (1, 0, \cdots, 0)$ and $\boldsymbol{v} = (\frac{1}{\sqrt{p}}, \cdots, \frac{1}{\sqrt{p}})$, both of length p.

Vector $\boldsymbol{u}$ is sparse, because it contains mostly zeros.

Using the $\boldsymbol{L}_1$ norm, we have $||\boldsymbol{u}||_1 = \sum_{i=1}^{p} |u_i| = 1$ and $||\boldsymbol{v}||_1 = \sum_{i=1}^{p} |v_i| = \sqrt{p}$.

Using the $\boldsymbol{L}_2$ norm, we have $||\boldsymbol{u}||_2 = \sum_{i=1}^{p} {u_i}^2 = 1$ and $||\boldsymbol{v}||_2 = \sum_{i=1}^{p} {v_i}^2 = 1$.

The $\boldsymbol{L}_1$ penalty rewards the sparsity of $\boldsymbol{u}$; the $\boldsymbol{L}_2$ penalty makes no distinction.

Summary

Regression Analysis

Model Selection

Nicoleta Serban, Ph.D.

Professor

Stewart School of Industrial and Systems Engineering

Regularized Regression: Approaches

Georgia Tech

About This Lesson

Variable Standardization & Notation

For regularized regression, center each column's values at zero and rescale so that the sum of squares of each column's values is 1. That is,

- Rescale the values for each *j*-th predicting variable, x_j, $j=1,.., p$, as follows:

$$\frac{1}{n}\sum_{i=1}^{n} x_{ij} = 0$$

and

$$\frac{1}{n}\sum_{i=1}^{n} x_{ij}^2 = 1$$

- It is also recommended to rescale the response variable in the same way:

$\frac{1}{n}\sum_{i=1}^{n} y_i = 0$ and $\frac{1}{n}\sum_{i=1}^{n} y_i^2 = 1$

➔ **Use the original scale when fitting the selected model for interpretation of the regression coefficients.**

Georgia Tech

Ridge Regression

- Minimizes SSE plus the penalty term

$$SSE_\lambda(\boldsymbol{\beta}) = \sum_{i=1}^{n}\left(y_i - (\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip})\right)^2 + \lambda \sum_{j=1}^{p} {\beta_j}^2$$

- Provides closed-form estimate of regression coefficients ($\widehat{\boldsymbol{\beta}}$)

$$\widehat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{X} + \lambda\mathbf{I})^{-1}\boldsymbol{X}^T\boldsymbol{Y}$$

  - $\mathbf{I}$ is the identity matrix
- $\lambda = 0$ gives least squares estimate (low bias, high variance)
- $\lambda \rightarrow \infty$ gives $\widehat{\boldsymbol{\beta}} \rightarrow 0$ (high bias, low variance)
- Commonly used under multicollinearity
- Not used for model selection
  - Shrinks but does not “force” any $\hat{\beta}_j$ to equal 0

LASSO Regression

- **L**east **A**bsolute **S**hrinkage and **S**election **O**perator
- Normal Linear Regression minimizes

$$SSE_\lambda(\boldsymbol{\beta}) = \sum_{i=1}^{n}\left(y_i - (\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip})\right)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

- Generalized Linear Model minimizes

$$SSE_\lambda(\boldsymbol{\beta}) = -\ell(\beta_0, \cdots, \beta_p) + \lambda \sum_{j=1}^{p} |\beta_j|$$

  - $\ell(\boldsymbol{\beta})$ is the log-likelihood function
- Estimated regression coefficients
  - Must use numerical algorithms
  - No closed-form expression
- Used for model selection
  - Does "force" some $\hat{\beta}_j$ to equal 0

Georgia Tech

LASSO Regression

- **L**east **A**bsolute **S**hrinkage and **S**election **O**perator
- Normal Linear Regression minimizes

$$SSE_\lambda(\boldsymbol{\beta}) = \sum_{i=1}^{n}\left(y_i - (\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip})\right)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

- Generalized Linear Model minimizes

$$SSE_\lambda(\boldsymbol{\beta}) = -\ell(\beta_0, \cdots, \beta_p) + \lambda \sum_{j=1}^{p} |\beta_j|$$

  - $\ell(\boldsymbol{\beta})$ is the log-likelihood function
- Estimated regression coefficients
  - Must use numerical algorithms
  - No closed-form expression
- Used for model selection
  - Does "force" some $\hat{\beta}_j$ to equal 0

- LASSO performs estimation of regression coefficients and variable selection simultaneously.
- The regression coefficients obtained from LASSO are less efficient than those obtained from Ordinary Least Squares (OLS).

➔ **After using LASSO to select the model, use OLS to estimate the (final) regression coefficients.**

Choosing λ: Cross-Validation

Split the data $\{(x_{11}, \cdots, x_{1p}), y_1\}, \cdots, \{(x_{n1}, \cdots, x_{np}), y_n\}$ into two sets.

- **Training set**
  - Use to fit the penalized model
    - Given λ, estimate $\hat{\beta}_0, \hat{\beta}_1, \cdots, \hat{\beta}_p$
- **Testing/Validation set**
  - Use to evaluate performance of model obtained with training set
    - Estimate mean squared error (MSE) for normal regression
    - Estimate classification error rate for logistic regression
    - Estimate sum of squared deviances for Poisson regression
    - Generally, estimate a scoring rule depending on the regression problem

The process can be repeated for multiple λs.

Cross Validation: How to Split Data?

K-fold cross-validation (KCV)

- Divide data into K chunks of approximately equal size
- For a range of λ penalty values, e.g., $\lambda_1, \cdots, \lambda_B$, and for $k = 1$ to K
  - The training set consists of data without the k-th fold of data, and the testing set consists of the k-th fold
  - Given λ, fit a model on the training data and predict responses
  - Given λ, compute mean squared error or classification error rate for the k-th fold testing data
  - Given λ, after K folds have been processed, compute overall error (e.g., MSE or classification error) for that λ for all folds

➔ Select λ penalty providing minimum overall error

Ridge vs. LASSO Regression

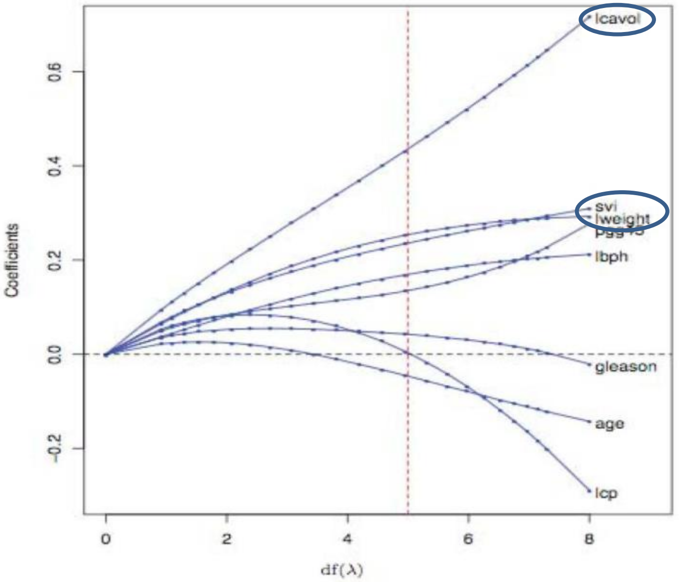


FIGURE 3.8. *Profiles of ridge coefficients for the prostate cancer example, as the tuning parameter λ is varied. Coefficients are plotted versus $\mathrm{df}(\lambda)$, the effective degrees of freedom. A vertical line is drawn at $\mathrm{df} = 5.0$, the value chosen by cross-validation.*

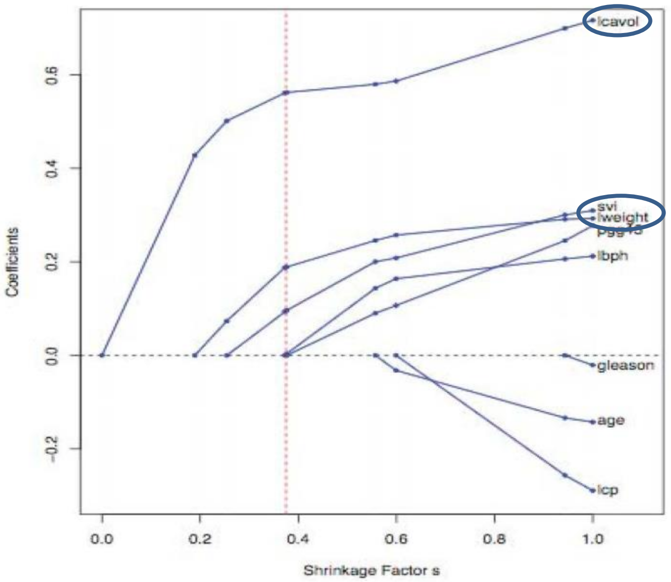


FIGURE 3.10. *Profiles of lasso coefficients, as the tuning parameter t is varied. Coefficients are plotted versus $s = t/\sum_1^p |\hat{\beta}_j|$. A vertical line is drawn at $s = 0.36$, the value chosen by cross-validation. Compare Figure 3.8 on page 65; the lasso profiles hit zero, while those for ridge do not. The profiles are piece-wise linear, and so are computed only at the points displayed; see Section 3.4.4 for details.*

Acknowledgement: From Hastie, T., Tibshirani, R., Friedman, J. (2001), *The Elements of Statistical Learning*, Springer Series in Statistics.

LASSO: Limitations

- LASSO selects only up to n variables
  - n is the number of observations
  - If the number of potential predictors is greater than the number of observations, LASSO will select at most n of them
  - Since, normally, $n > p$, not a significant limitation
- If there are high correlations among predictors
  - LASSO is dominated by ridge regression
- If there is a group of variables with high correlation
  - LASSO tends to select only one variable from the group
    - LASSO doesn't care which one

Elastic Net

Elastic Net minimizes

$$\sum_{i=1}^{n}\left(y_i - (\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip})\right)^2 + \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{j=1}^{p} {\beta_j}^2$$

- $\boldsymbol{L_1}$ penalty generates a sparse model
- $\boldsymbol{L_2}$ penalty
  - Removes the limitation on the number of selected variables
  - Encourages group effect
  - Stabilizes the $\boldsymbol{L_1}$ regularization path

Reference: Hui Zou and Trevor Hastie. "Regularization and variable selection via the elastic net." *Journal of the Royal Statistical Society: Series B* 67.2 (2005): 301-320.

Georgia Tech

Summary

Regression Analysis

Model Selection

Nicoleta Serban, Ph.D.

Professor

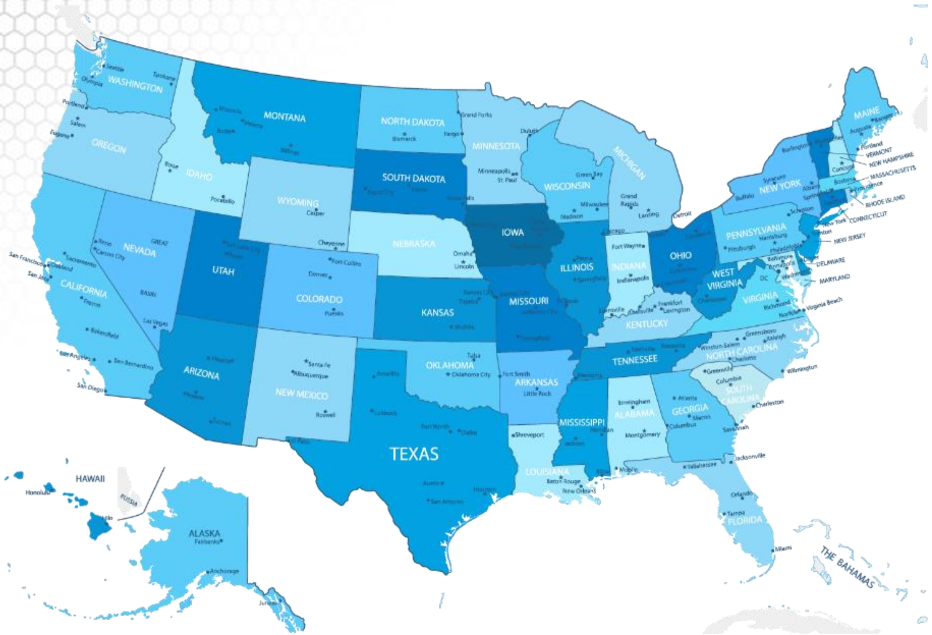
Stewart School of Industrial and Systems Engineering

Regularized Regression:
Data Examples

Georgia Tech

About This Lesson

Ranking States by SAT Performance



SAT Mean Score by State – Year 1982
790 (South Carolina) – 1088 (Iowa)

- ***Which variables are associated with state average SAT scores?***
- ***After accounting for selection biases, how do the states rank?***
- ***Which states perform best for the amount of money they spend?***

Ridge Regression

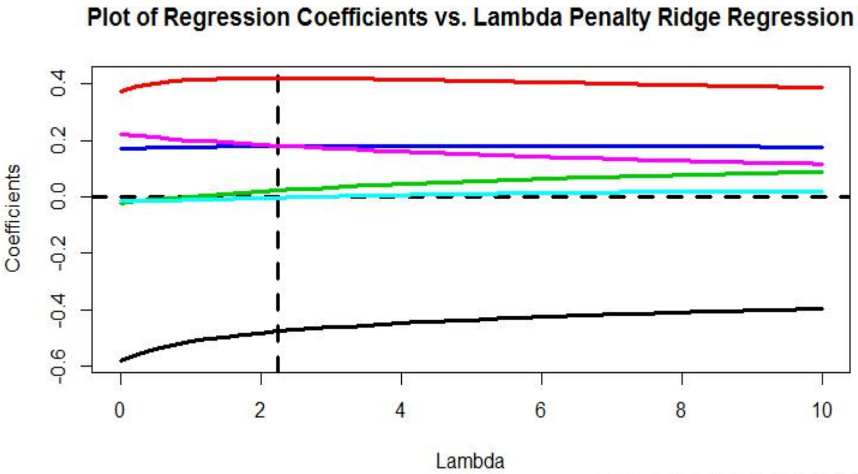
```
library(MASS)
## Scale the predicting variables and the response variable
ltakers = log(takers)
predictors = cbind(ltakers, rank, income, years, public, expend)
predictors = scale(predictors)
sat.scaled = scale(sat)

## Apply ridge regression for a range of penalty constants
lambda = seq(0, 10, by=0.25)
out = lm.ridge(sat.scaled~predictors, lambda=lambda)
round(out$GCV, 5)

which(out$GCV == min(out$GCV))
 2.25
    10
round(out$coef[,10], 4)
predictorsltakers    predictorsrank  predictorsincome   predictorsyears  predictorspublic
          -0.4771            0.4195            0.0223            0.1796           -0.0028
 predictorsexpend
           0.1808
```

The ridge regression outputs estimates for each lambda in the considered range *(not shown)*

The lambda is selected to minimize the (generalized) CV score

Ridge Regression

```
plot(lambda, out$coef[1,], type = "l", col=1, lwd=3,
         xlab = "Lambda", ylab = "Coefficients",
         main = "Plot of Regression Coefficients vs. Lambda
Penalty Ridge Regression",
         ylim = c(min(out$coef), max(out$coef)))
for(i in 2:6)
    points(lambda, out$coef[i,], type = "l", col=i, lwd=3)
abline(h = 0, lty = 2, lwd = 3)
abline(v = 2.25, lty = 2, lwd=3)
```



LASSO Regression

```
library(lars)
object = lars(x=predictors, y=sat.scaled)
Object
```

Sequence of LASSO moves:

	ltakers	rank	years	expend	income	public
Var	1	2	4	6	3	5
Step	1	2	3	4	5	6

```
round(object$Cp,2)
```

0	1	2	3	4	5	6
349.91	103.40	46.89	35.64	3.10	5.09	7.00

The selected model according to Malow's Cp is at the fourth variable introduced in the model.

LASSO Regression: First Implementation

library(lars)
object = lars(x=predictors, y=sat.scaled)
Object

Sequence of LASSO moves:

	ltakers	rank	years	expend	income	public
Var	1	2	4	6	3	5
Step	1	2	3	4	5	6

round(object$Cp,2)

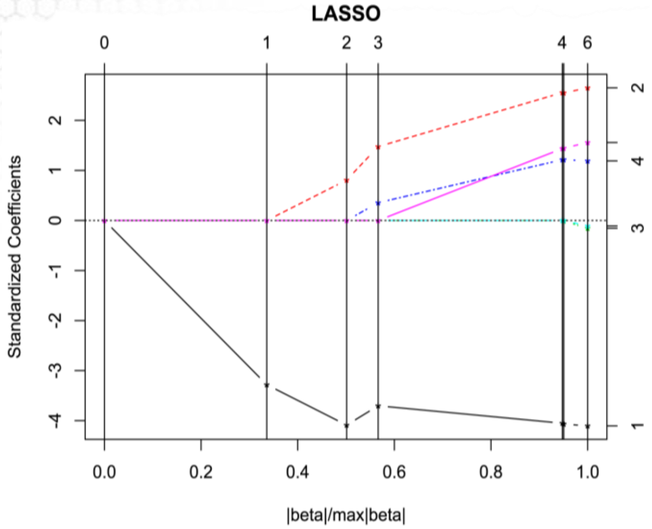
0	1	2	3	4	5	6
349.91	103.40	46.89	35.64	3.10	5.09	7.00

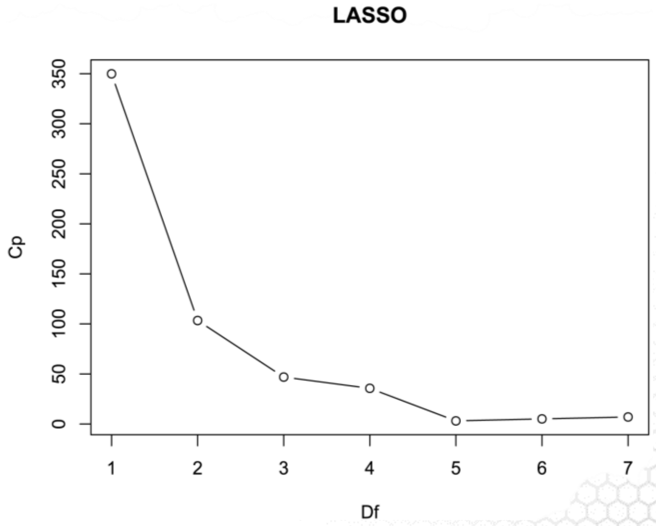
The selected model according to Malow's Cp is at the fourth variable introduced in the model.

- From the order the predictors were added, i.e., log(*takers*), *rank*, *years*, *expend*, *income* and *public*, the first four are selected
- After LASSO variable selection, apply ordinary least squares (OLS) with the selected predicting variables

LASSO Regression: First Implementation

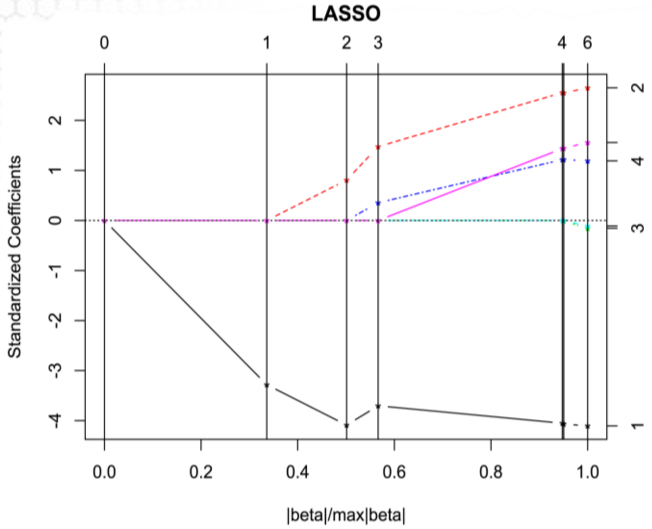
plot.lars(object)
plot.lars(object, xvar="df", plottype="Cp")

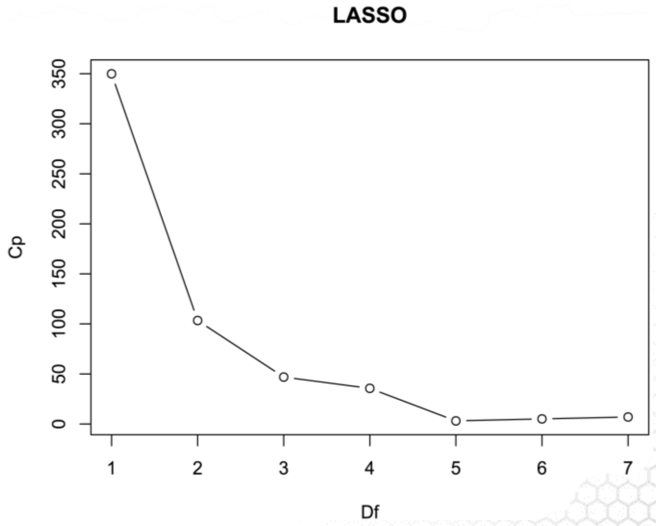




LASSO Regression: First Implementation

plot.lars(object)
plot.lars(object, xvar="df", plottype="Cp")





From the order the predictors were added, i.e., log(*takers*), *rank*, *years*, *expend*, *income* and *public*, the first four are selected

LASSO Regression: Second Implementation

```
library(glmnet) # alpha=1 lasso, alpha=0 ridge
Xpred= cbind(ltakers, rank, income, years, public, expend)

# Find the optimal lambda using 10-fold CV
satmodel.cv=cv.glmnet(Xpred, sat, alpha=1, nfolds=10)

## Fit lasso model with 100 values for lambda
satmodel = glmnet(Xpred, sat, alpha = 1, nlambda=100)

## Extract coefficients at optimal lambda
coef(satmodel, s=satmodel.cv$lambda.min)
```

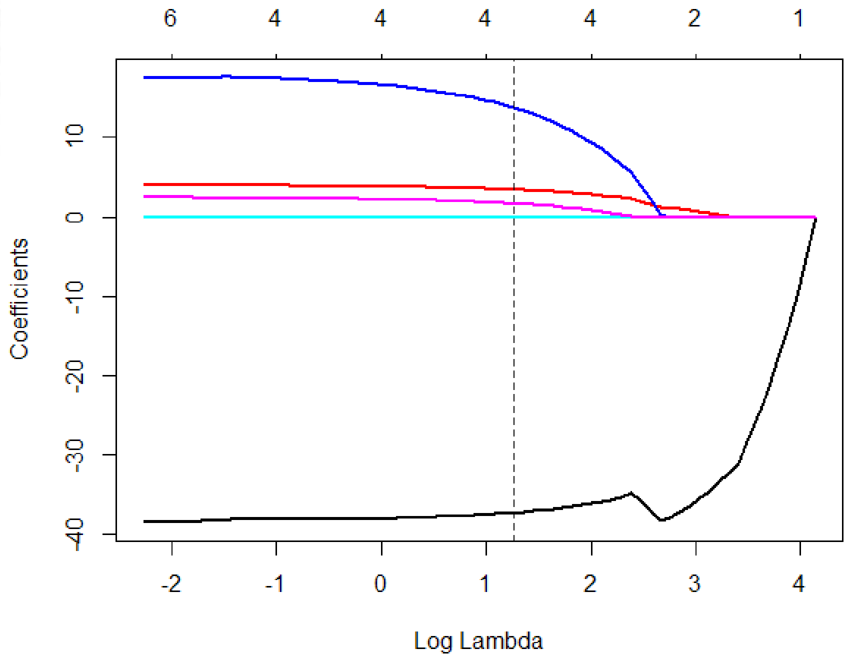
(Intercept)	516.519096
ltakers	-37.244873
rank	3.420034
income	.
years	13.780563
public	.
expend	1.662338

Because CV uses random assignments, expect slightly different coefficients each time it is run.

Using LASSO and the penalty selected using 10-fold CV, the selected predictors are: log(*takers*), *rank*, *years*, and *expend*

LASSO Regression: Second Implementation

Plot coefficient paths
plot(satmodel,xvar="lambda", lwd=2)
abline(v=log(satmodel.cv$lambda.min), col='black', lty=2)



Elastic Net Regression

```
library(glmnet) # alpha=1 lasso, alpha=0 ridge
Xpred= cbind(ltakers, rank, income, years, public, expend)

# Find the optimal lambda using 10-fold CV
satmodel.cv=cv.glmnet(Xpred, sat, alpha=0.5, nfolds=10)

## Fit elastic net model with 100 values for lambda
satmodel = glmnet(Xpred, sat, alpha=0.5, nlambda = 100)

## Plot coefficient paths
coef(satmodel, s=satmodel.cv$lambda.min)
```

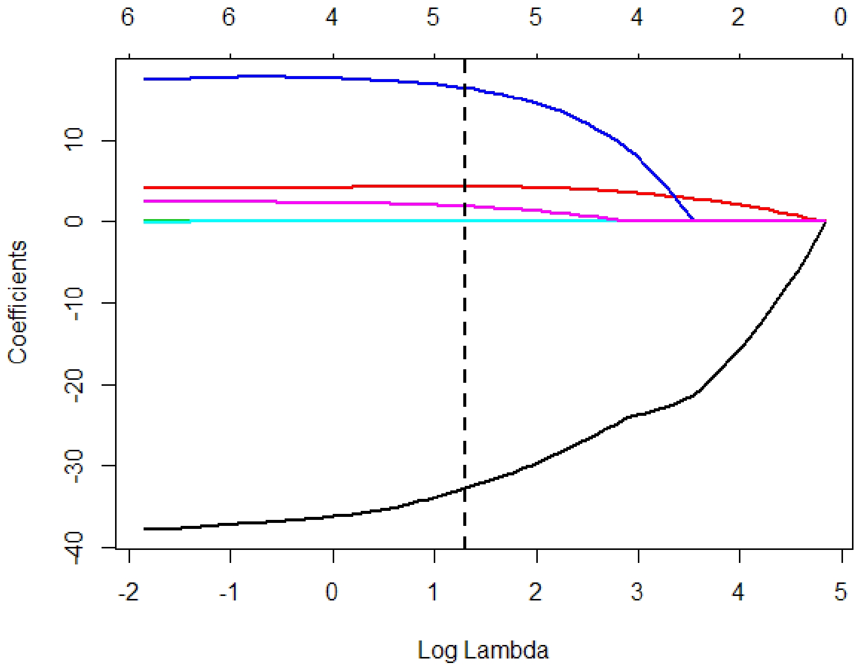
(Intercept)	386.70798566
ltakers	-32.76143283
rank	4.24246653
income	0.01727723
years	16.40041447
public	.
expend	1.83649835

Because CV uses random assignments, expect slightly different coefficients each time it is run.

Using Elastic Net and the penalty selected using 10-fold CV, the selected predictors are: log(*takers*), *rank*, *income*, *years*, and *expend*

Elastic Net

Extract coefficients at optimal lambda
plot(satmodel, xvar="lambda", lwd=2)
abline(v=log(satmodel.cv$lambda.min), col='black', lty=2, lwd=2)



Overview of All Selection Approaches

	Log(Takers)	Rank	Income	Years	Public	Expend
Best Subset & Mallow's Cp		×		×	×	×
Stepwise & AIC	×	×		×		×
LASSO & Mallow's Cp	×	×		×		×
Lasso & 10-fold CV	×	×		×		×
Elastic Net & 10-fold CV	×	×	×	×		×

- *Rank*, *Years*, and *Expend* are selected by all approaches
- Best Subset alone selects *Public* and does not select *Takers*
- *Income* is selected only by Elastic Net

Georgia Tech

Predicting Bankruptcy

- Effective bankruptcy prediction is useful for investors and analysts, allowing for accurate evaluation of a firm's prospects.

- Roughly 40 years ago, Ed Altman showed that publicly available financial indicators can be used to distinguish between firms that are about to go bankrupt and those that are not.

Which financial indicators are associated with bankruptcy for telecommunications firms?

Acknowledgement: This example was provided by Dr. Jeffrey Simonoff from New York University and was inspired by the honors thesis of Jeffrey Lui.

Georgia Tech

LASSO Regression

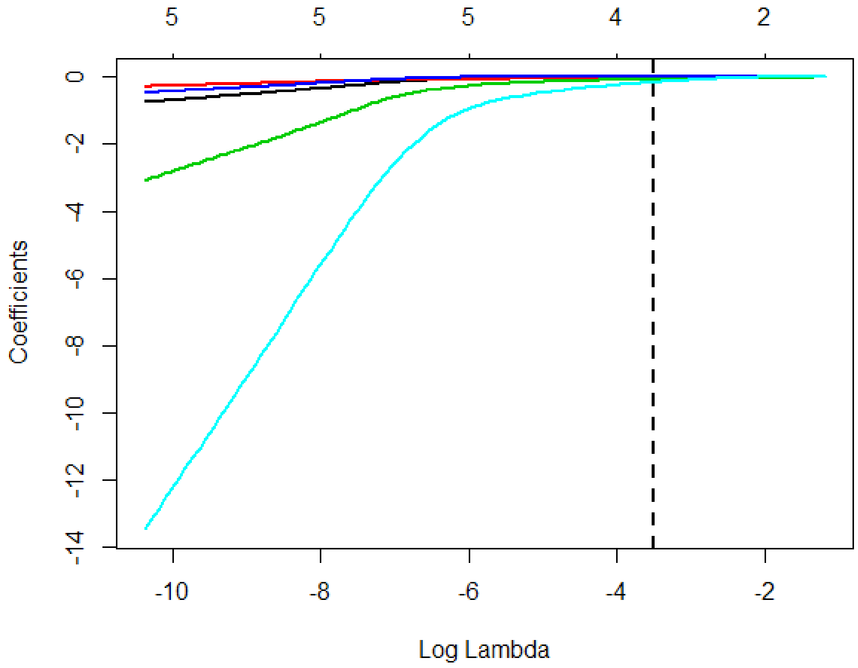
```
library(glmnet)
X = cbind(WC.TA, RE.TA, EBIT.TA, S.TA, BVE.BVL)

## 10-fold CV to find the optimal lambda
bank5.cv = cv.glmnet(X, Bankrupt, family=c("binomial"), alpha=1, type="class", nfolds=10)

## Fit lasso model with 100 values for lambda
bank5 = glmnet(X, Bankrupt, family=c("binomial"), alpha=1, nlambda=100)

## Extract coefficients at optimal lambda
coef(bank5, s=bank5.cv$lambda.min)
```

(Intercept)	-0.95995368
WC.TA	.
RE.TA	-0.02874387
EBIT.TA	-0.05757731
S.TA	.
BVE.BVL	-0.14135425

Using LASSO and the penalty selected using 10-fold CV, the selected predictors are: *RE.TA*, *EBIT.TA*, and *BE.BVL*

LASSO Regression

Plot coefficient paths
plot(bank5, xvar="lambda", lwd=2)
abline(v=log(bank5.cv$lambda.min), col='black', lty 2, lwd=2)



Elastic Net Regression

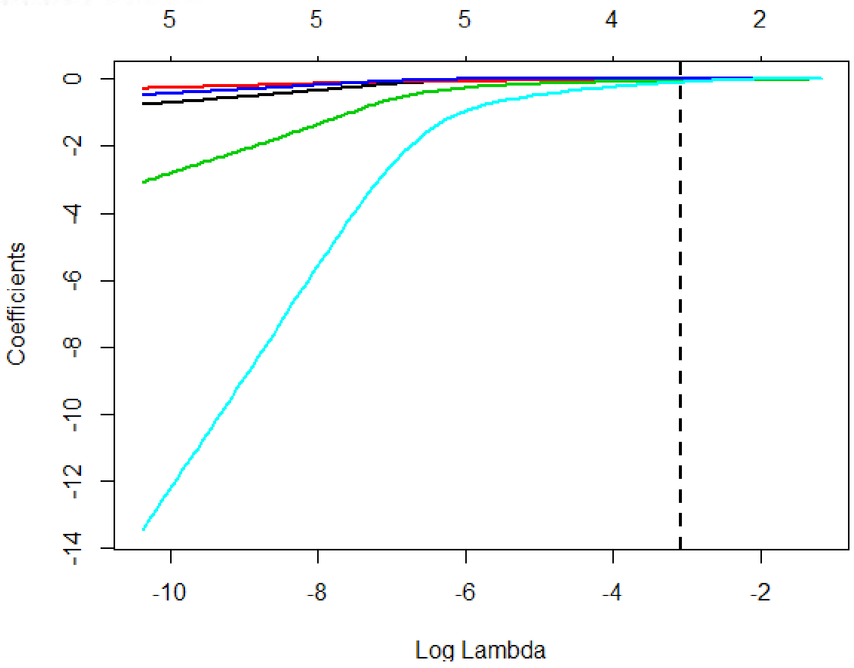
```
library(glmnet) # alpha=1 lasso, alpha=0 ridge
X = cbind(WC.TA,RE.TA,EBIT.TA,S.TA,BVE.BVL)

## 10-fold CV to find the optimal lambda
bank6.cv = cv.glmnet(X, Bankrupt, family=c("binomial"), alpha= 0.5, type="class", nfolds=10)

## Fit elastic net model with 100 values for lambda
bank6 = glmnet(X, Bankrupt, family=c("binomial"), alpha=0.5, nlambda=100)

## Extract coefficients at optimal lambda
coef(bank6, s=bank6.cv$lambda.min)
```

(Intercept)	-0.572208580
WC.TA	-0.006693551
RE.TA	-0.015677213
EBIT.TA	-0.050962740
S.TA	.
BVE.BVL	-0.108940580

Using Elastic Net and the penalty selected using 10-fold CV, the selected predictors are: *WC.TA*, *RE.TA*, *EBIT.TA*, and *BVE.BVL*

Elastic Net

Plot coefficient paths
plot(bank6, xvar="lambda", lwd=2)
abline(v=log(bank6.cv$lambda.min), col='black', lty=2, lwd=2)



Overview of All Selection Approaches

	WC.TA	RE.TA	EBIT.TA	S.TA	BVE.BVL
Best subset AIC		×	×		×
Stepwise & AIC		×	×		×
Lasso & 10-fold CV		×	×		×
Elastic Net & 10-fold CV	×	×	×		×

Summary

Regression Analysis

Regression Analysis in Practice

Nicoleta Serban, Ph.D.
Professor
Stewart School of Industrial and Systems Engineering

Emergency Department
Healthcare Costs

About This Lesson

Emergency Department Healthcare Costs

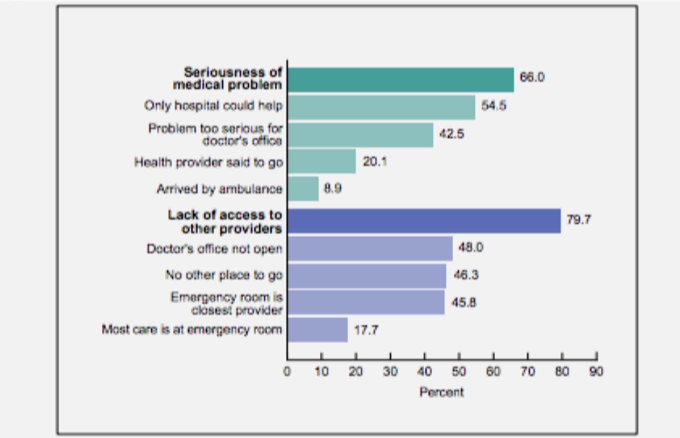


Figure 1. Percentage who had selected reasons for last emergency room visit, among adults aged 18–64 whose last visit in past 12 months did not result in hospital admission: United States, January–June 2011

Research Question 1
What factors impact the healthcare cost due to emergency department encounters?

Research Question 2
Is access to primary care providers associated with healthcare costs due to emergency department encounters?

Emergency Department Healthcare Costs

Study Population

Adults enrolled in Medicaid in 2011 in four southeast states

- Alabama, Arkansas, Louisiana, and North Carolina
- Medicaid is a low-income health insurance program

Data Source

Medicaid Analytic eXtract (MAX) claims files available from the Centers of Medicare and Medicaid Services (CMS)

- Additional data sources: U.S. Bureau Census, Health Analytics Group at GT, Robert Wood Johnson Foundation, among others.
- Disclaimer: This analysis of healthcare cost for the Medicaid population using the MAX claims data is in compliance with the study protocol approved by the Georgia Tech Internal Review Board (IRB) and by CMS. Do NOT use the data provided for this analysis for purposes other than the study in this lesson.

Georgia Tech

Response Variable

- *EDcost*
  - Primary variable of interest
  - Emergency Department cost aggregated at the census tract level
  - Depends on number of enrollees (members) and lengths of their enrollments

- *PMPM*
  - Per Member Per Month
  - Total number of enrollment months aggregated by census tract
  - Used to scale *EDcost* for comparison across census tracts
    - Each census tract has different numbers of enrollees, and each enrollee can have a different length of enrollment
    - Scaling *EDcost* by *PMPM* allows a comparison of cost per enrollee month

Georgia Tech

Predicting Variables

- **Location**
  - *State* and *GEOID* give state and census tract identification
- **Utilization**
  - Data must be scaled by *PMPM*
    - *ED* (number of emergency department claims)
    - *HO* (number of hospitalization claims)
    - *PO* (number of physician office claims)
- **Population characteristics**
  - Percentages of Medicaid-enrolled adults of various populations
    - *BlackPop*, *WhitePop*, *OtherPop* (race/ethnicity)
    - *HealthyPop*, *ChronicPop, ComplexPop* (health conditions)
- **Socioeconomic and Health Environment Factors**
  - 13 variables quantifying other possibly health-related factors
    - Includes unemployment, median income, urbanicity of the census tract, access to primary care, health rankings, and others

Controlling Variables

Selection Bias

- Adults with chronic or complex health problems tend to need emergency services more than the healthy population
- Controlling factors
  - *ChronicPop* (percentage of population with chronic conditions)
  - *ComplexPop* (percentage with complex health problems)

Confounding Variable

- The number of ED claims correlates with both response and predicting variables
  - It is a measure of the utilization of the emergency department
    - Utilization directly leads to ED healthcare costs
    - It is therefore a confounding variable
    - Do not include this confounding variable in the model

Summary

Regression Analysis

Regression Analysis in Practice

Nicoleta Serban, Ph.D.

Professor

Stewart School of Industrial and Systems Engineering

Emergency Department Healthcare Costs: Exploratory Data Analysis

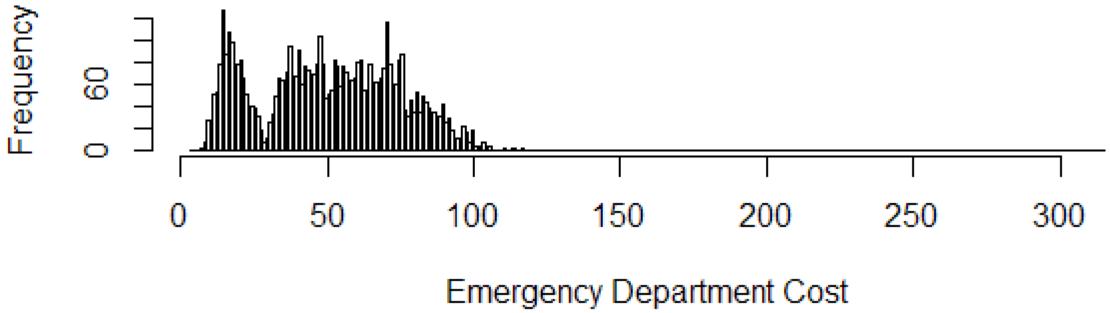
Georgia Tech

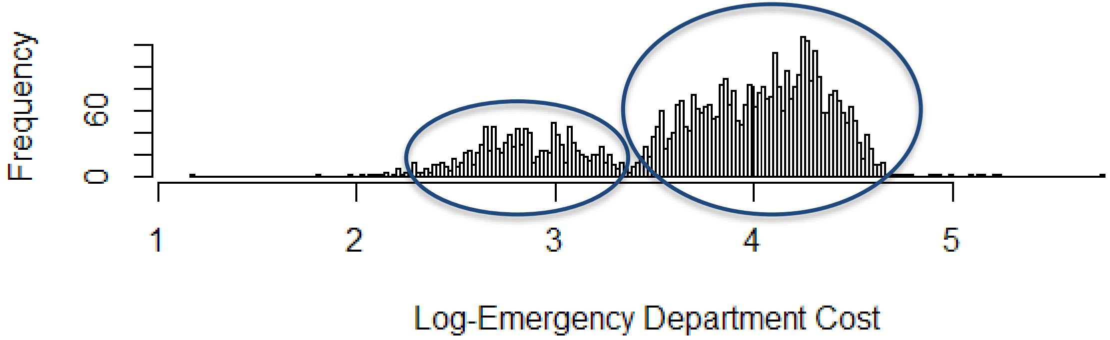
About This Lesson

Exploratory Data Analysis: Response Variable

```
## Read the data using read.csv() R command
dataAdult = read.csv("DataADULT.csv", header=TRUE)
attach(dataAdult)

## Rescale outcome/response variable
EDCost.pmpm = EDCost/PMPM

## Rescale utilization
dataAdult$PO = PO/PMPM
dataAdult$HO = HO/PMPM

## Histogram of the response variable
par(mfrow=c(2,1))
hist(EDCost.pmpm, breaks=300, xlab="Emergency Department Cost", main="")
hist(log(EDCost.pmpm), breaks=300, xlab="Log-Emergency Department Cost", main="")
```

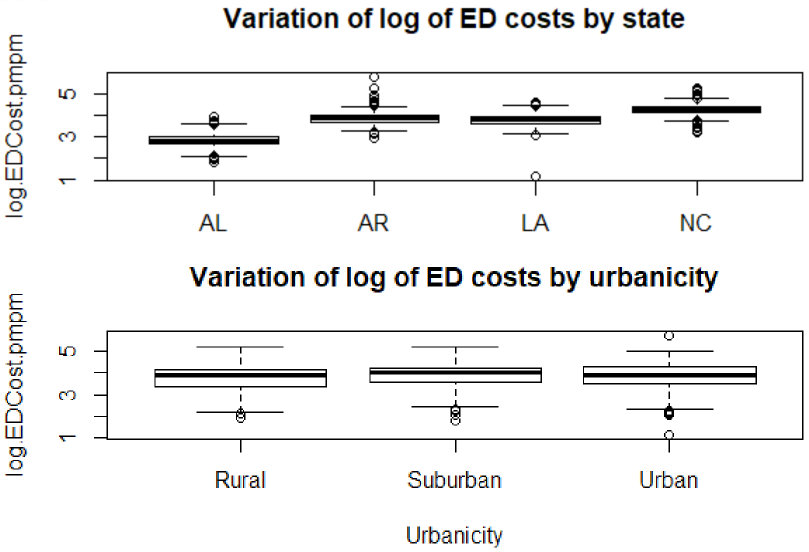
Exploratory Data Analysis: Response Variable





Exploratory Data Analysis: Response vs Qualitative Predictors

```
log.EDCost.pmpm = log(EDCost.pmpm)
## Response variable vs categorical predicating variables
par(mfrow=c(2,1))
boxplot(log.EDCost.pmpm ~ State, main = "Variation of log of ED costs by state")
boxplot(log.EDCost.pmpm ~ Urbanicity, main = "Variation of log of ED costs by urbanicity")
```

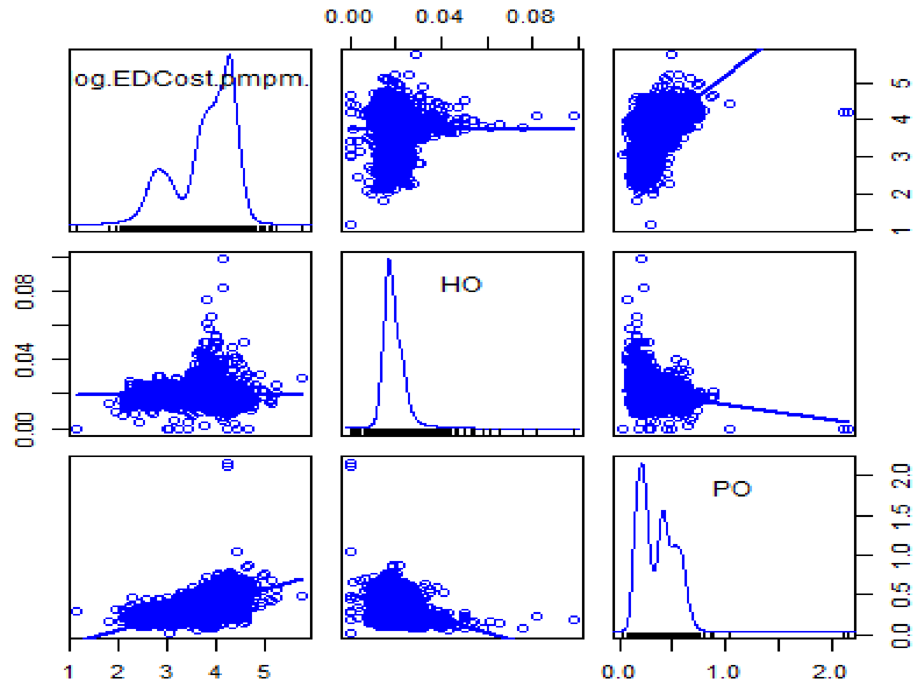


Exploratory Data Analysis: Response vs Quantitative Predictors

```
## Scatterplot matrix plots
library(car)

## Response vs Utilization
scatterplotMatrix(~ log(EDCost.pmpm) + HO + PO, smooth=FALSE)

## Response vs Population Characteristics
scatterplotMatrix(~ log(EDCost.pmpm) + WhitePop + BlackPop + OtherPop + HealthyPop +
        ChronicPop + ComplexPop, smooth=FALSE)

## Response vs Socioeconomic and Environmental Characteristics
scatterplotMatrix(~ log(EDCost.pmpm) + Unemployment + Income + Poverty + Education +
        Accessibility + Availability + ProvDensity, smooth=FALSE)

## Response vs County Health Rankings
scatterplotMatrix(~ log(EDCost.pmpm) + RankingsPCP + RankingsFood + RankingsHousing +
        RankingsExercise + RankingsSocial, smooth=FALSE)
```

Response vs Quantitative Predictors

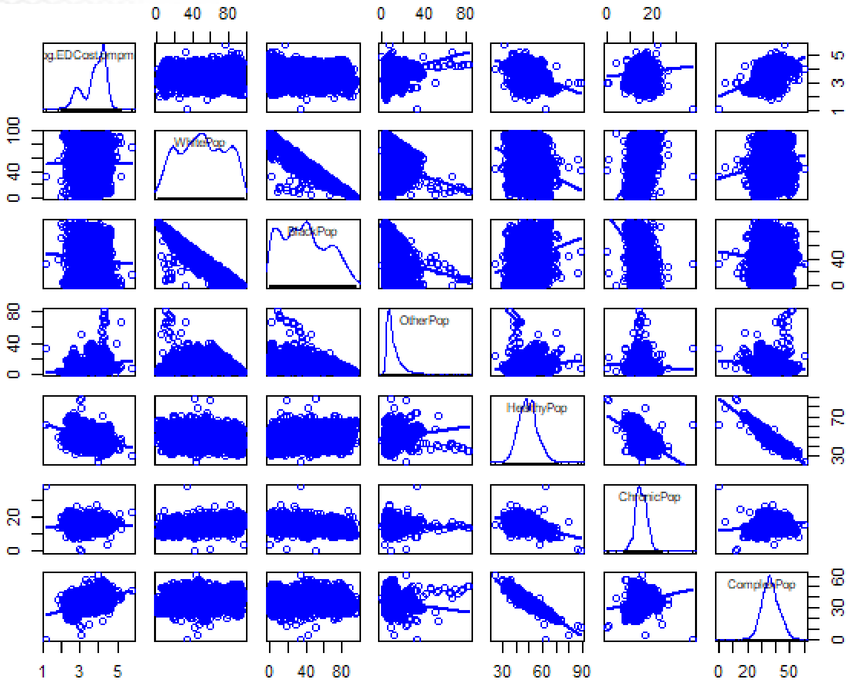
ED Cost vs. Utilization Measures: *Number of Claims for HO and PO*



Georgia Tech

Response vs Quantitative Predictors

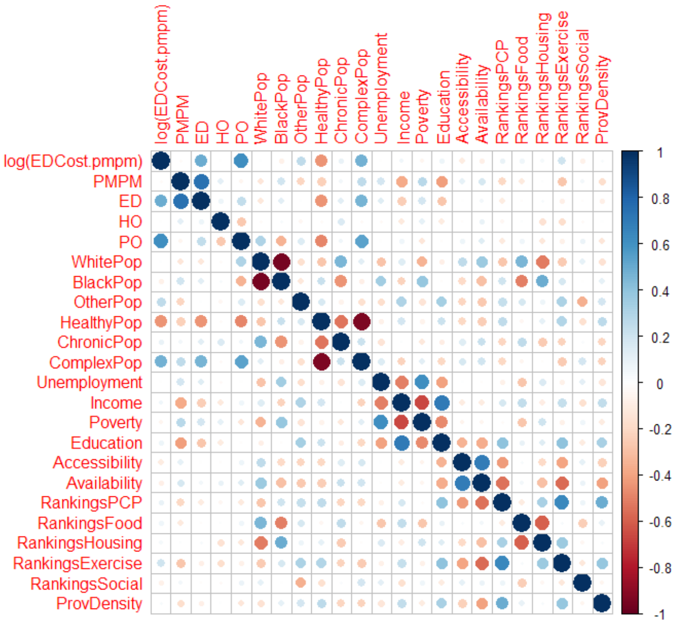
ED Cost vs. Population Characteristics: *WhitePop, BlackPop, OtherPop, HealthyPop, ChronicPop, ComplexPop*



Georgia Tech

Response vs. Predicting Variables: Correlation Matrix Plot

```
## Correlation matrix plot
library(corrplot)
corr = cor(cbind(log(EDCost.pmpm), dataAdult[,-c(1, 2, 3, 18)]))
corrplot(corr)
```



Summary

Regression Analysis

Regression Analysis in Practice

Nicoleta Serban, Ph.D.
Professor
Stewart School of Industrial and Systems Engineering

Emergency Department Healthcare Costs : Model Fit and Assessment

Georgia Tech

About This Lesson

Multiple Linear Regression Model

```
## Exclude GEOID, scaling factor PMPM, and confounding factors EDCost and ED
## Exclude OtherPop & ComplexPop because of linear dependence
dataAdult.red = dataAdult[, -c(1, 3, 4, 5, 10, 13)]

fullmodel = lm(log(EDCost.pmpm) ~ ., data=dataAdult.red)
summary(fullmodel)
```

Multiple Linear Regression Model

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	2.208e+00	1.175e-01	18.788	< 2e-16 ***
StateAR	9.235e-01	1.610e-02	57.353	< 2e-16 ***
StateLA	9.081e-01	1.358e-02	66.853	< 2e-16 ***
StateNC	1.418e+00	1.650e-02	85.909	< 2e-16 ***
HO	1.168e+01	7.587e-01	15.401	< 2e-16 ***
PO	1.378e-01	4.114e-02	3.350	0.000815 ***
WhitePop	4.416e-03	5.800e-04	7.614	3.16e-14 ***
BlackPop	4.894e-03	5.824e-04	8.403	< 2e-16 ***
HealthyPop	-9.044e-04	8.160e-04	-1.108	0.267751
ChronicPop	-5.949e-03	2.052e-03	-2.899	0.003760 **
Unemployment	4.390e-04	7.377e-04	0.595	0.551797
Income	-2.556e-07	2.774e-07	-0.922	0.356769
Poverty	-3.306e-04	4.460e-04	-0.741	0.458529
Education	-1.447e-03	3.296e-04	-4.390	1.16e-05 ***
UrbanicitySuburban	-4.565e-04	1.369e-02	-0.033	0.973406
UrbanicityUrban	2.067e-02	1.269e-02	1.629	0.103356
Accessibility	-1.965e-03	7.094e-04	-2.770	0.005623 **
Availability	8.037e-02	1.975e-02	4.068	4.81e-05 ***
RankingsPCP	7.596e-04	1.819e-04	4.175	3.03e-05 ***
RankingsFood	6.586e-03	5.203e-03	1.266	0.205642
RankingsHousing	-4.642e-03	1.562e-03	-2.973	0.002967 **
RankingsExercise	3.993e-04	2.332e-04	1.712	0.086907 .
RankingsSocial	-3.895e-04	1.347e-03	-0.289	0.772497
ProvDensity	6.042e-02	1.573e-02	3.841	0.000124 ***

Socioeconomic predicting variables *Unemployment*, *Income*, *Poverty* and *RankingsSocial* are **not** statistically significant given other predicting variables in the model.

Access to primary care variables *Accessibility* and *Availability* are statistically significant.

85% of the variability in the ED cost is explained.

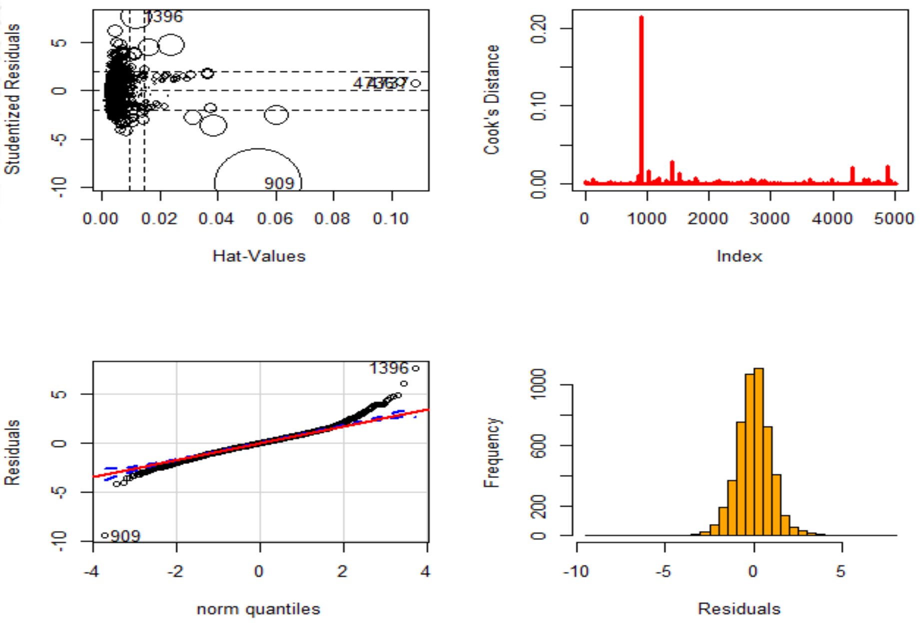
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2321 on 4995 degrees of freedom
Multiple R-squared: 0.8486,	Adjusted R-squared: 0.8479
F-statistic: 1218 on 23 and 4995 DF, p-value: < 2.2e-16

Residual Analysis: Outliers & Normality

```
## Residuals versus individual predicting variables
full.resid = rstandard(fullmodel)
cook = cooks.distance(fullmodel)

## Check outliers
influencePlot(fullmodel)
plot(cook, type="h", lwd=3, col="red", ylab="Cook's Distance")

## Check Normality
qqPlot(full.resid, ylab="Residuals", main = "")
qqline(full.resid, col="red", lwd=2)
hist(full.resid, xlab="Residuals", main = "", nclass=30, col="orange")
```

Residual Analysis: Outliers & Normality



Outliers

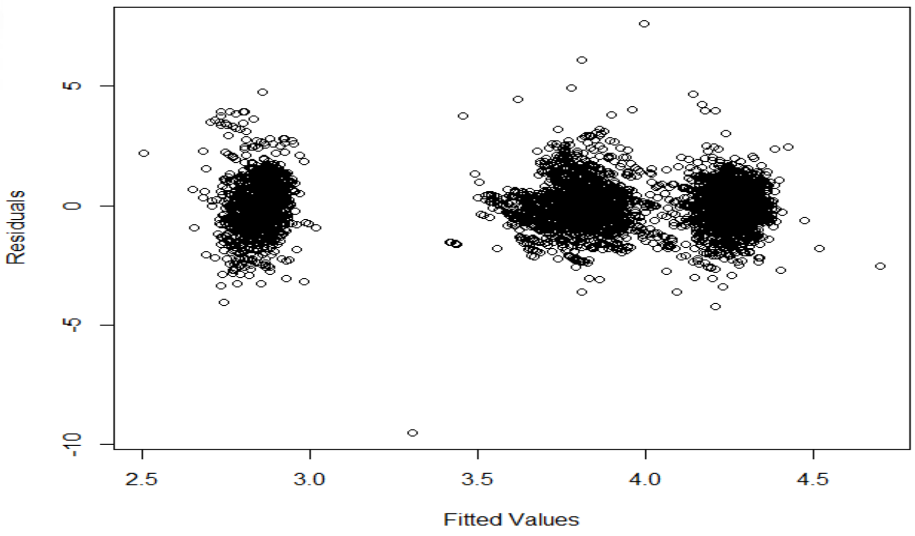
Observation 909 stands out.

Normality

Symmetric, but with heavy tails.

Residual Analysis: Constant Variance and Uncorrelated Errors

```
## Check Constant Variance & Uncorrelated Errors
full.fitted = fitted(fullmodel)
par(mfrow=c(1,1))
plot(full.fitted, full.resid, xlab="Fitted Values", ylab="Residuals")
```



Constant Variance Assumption

No pattern

Uncorrelated Errors Assumption

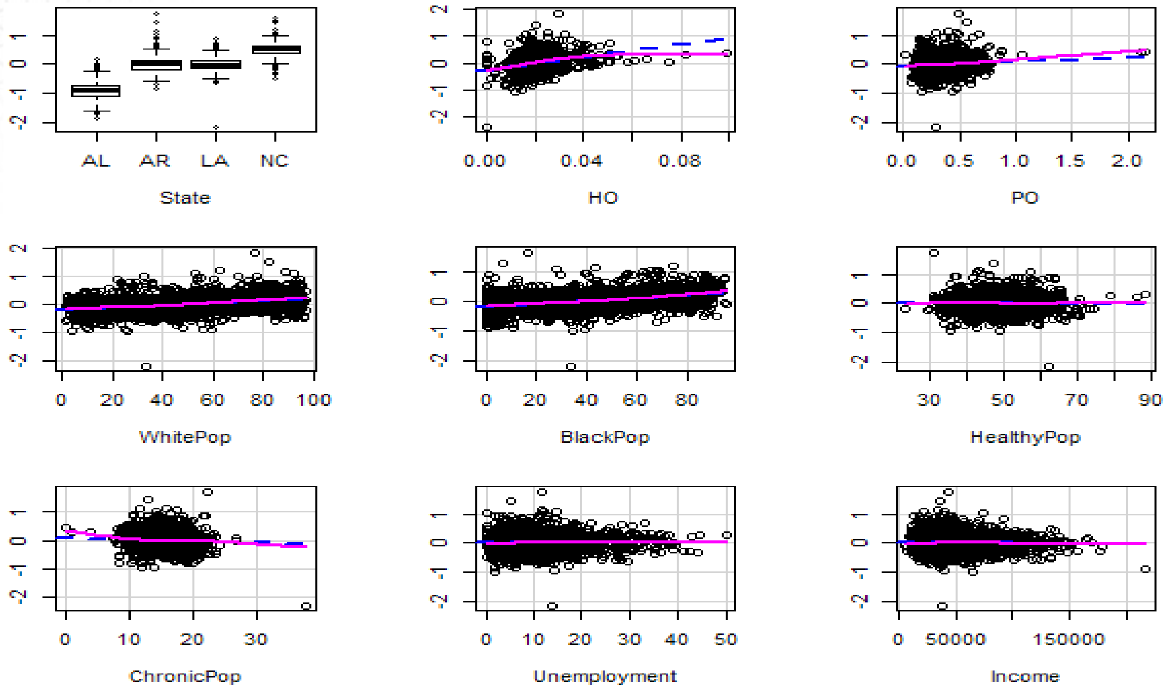
Three well-defined clusters

Spatial Dependence

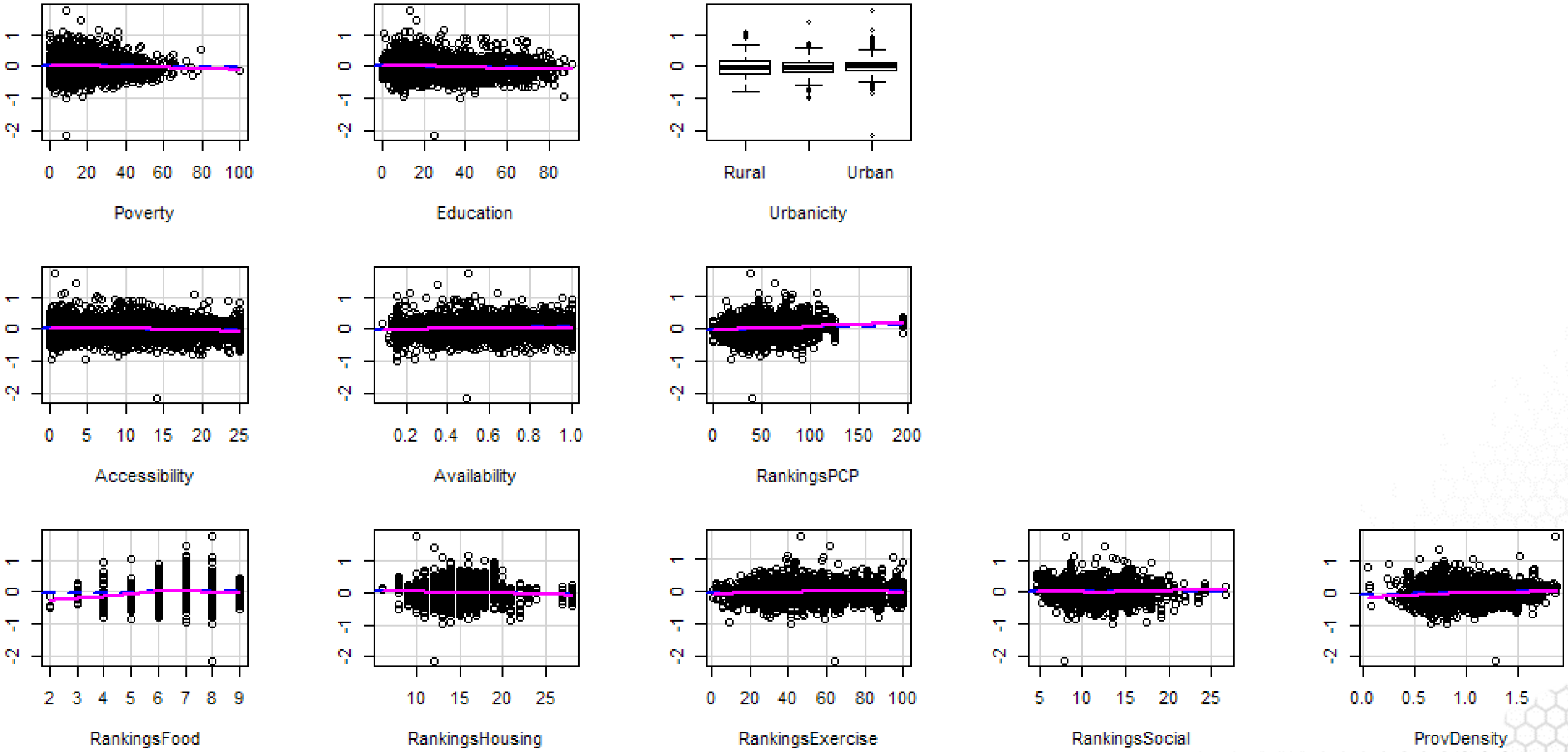
Residual Analysis: Linearity

Check Linearity

crPlots(fullmodel, ylab="")



Residual Analysis: Linearity (cont'd)



Georgia Tech

Summary

Regression Analysis

Regression Analysis in Practice

Nicoleta Serban, Ph.D.

Professor

Stewart School of Industrial and Systems Engineering

Emergency Department Healthcare Costs: Variable Selection

About This Lesson

Lasso Regression

```
predictors = as.matrix(dataAdult[, -c(1, 2, 3, 4, 5, 10, 13, 18)])

# Set up indicator (dummy) variables for State and Urbanicity
# Leave out one indicator (dummy) variable for each group

#AL= rep(0, length(State))
AR = rep(0, length(State))
LA = rep(0, length(State))
NC = rep(0, length(State))
#AL[as.numeric(factor(State))==1] = 1
 AR[as.numeric(factor(State))==2] = 1
 LA[as.numeric(factor(State))==3] = 1
 NC[as.numeric(factor(State))==4] = 1

#rural    = rep(0, length(Urbanicity))
suburban = rep(0, length(Urbanicity))
urban     = rep(0, length(Urbanicity))
#  rural[as.numeric(factor(Urbanicity))==1] = 1
suburban[as.numeric(factor(Urbanicity))==2] = 1
   urban[as.numeric(factor(Urbanicity))==3] = 1

predictors = cbind(predictors, AR, LA, NC, suburban, urban)
```

Lasso Regression

```r
## 10-fold CV to find the optimal lambda
lassomodel.cv = cv.glmnet(predictors, log(EDCost.pmpm), alpha=1, nfolds=10)

## Fit lasso model with 100 values for lambda
lassomodel = glmnet(predictors, log(EDCost.pmpm), alpha=1, nlambda=100)

## Plot coefficient paths
plot(lassomodel, xvar="lambda", label=TRUE, lwd=2)
abline(v=log(lassomodel.cv$lambda.min), col='black', lty=2, lwd=2)

## Extract coefficients at optimal lambda
coef(lassomodel, lassomodel.cv$lambda.min)
```

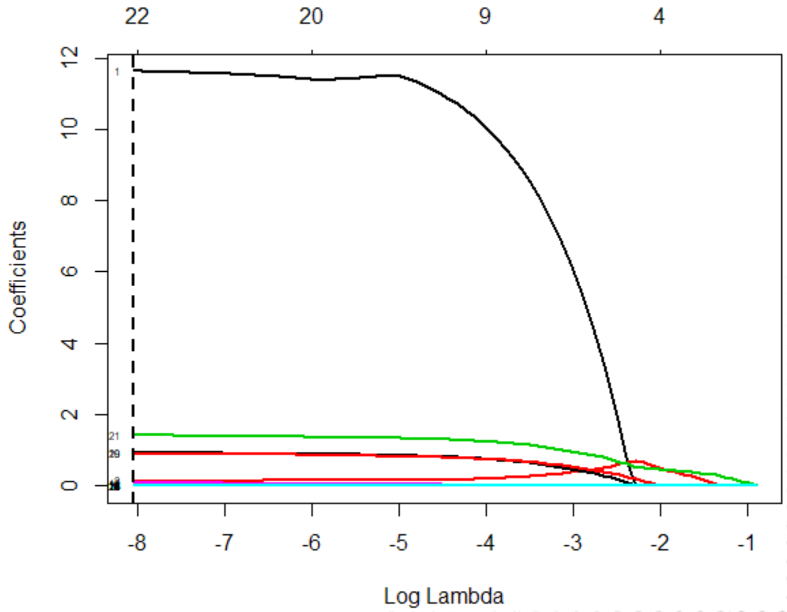
Lasso Regression

(Intercept)	2.277008e+00
HO	1.162649e+01
PO	1.389343e-01
WhitePop	3.767074e-03
BlackPop	4.246413e-03
HealthyPop	-1.042170e-03
ChronicPop	-5.704991e-03
Unemployment	3.421637e-04
Income	-2.307290e-07
Poverty	-2.383079e-04
Education	-1.451700e-03
Accessibility	-1.831102e-03
Availability	7.664592e-02
RankingsPCP	7.194696e-04
RankingsFood	5.782113e-03
RankingsHousing	-4.587208e-03
RankingsExercise	3.969711e-04
RankingsSocial	.
ProvDensity	5.923880e-02
AR	9.183680e-01
LA	9.027530e-01
NC	1.410464e+00
suburban	-7.302043e-05
urban	2.096038e-02

High-coefficient path corresponds to *HO* variable

RankingsSocial dummy variable is not selected

Other large-coefficient paths correspond to State dummy variables (*AR*, *LA*, *NC*)



Georgia Tech

Elastic Net Regression

```
## 10-fold CV to find the optimal lambda
enetmodel.cv = cv.glmnet(predictors, log(EDCost.pmpm), alpha=0.5, nfolds=10)

## Fit elastic net model with 100 values for lambda
enetmodel = glmnet(predictors, log(EDCost.pmpm), alpha=0.5, nlambda=100)

## Plot coefficient paths
plot(enetmodel, xvar="lambda", label=TRUE, lwd=2)
abline(v=log(enetmodel.cv$lambda.min), col='black', lty=2, lwd=2)

## Extract coefficients at optimal lambda
coef(enetmodel, s=enetmodel.cv$lambda.min)
```

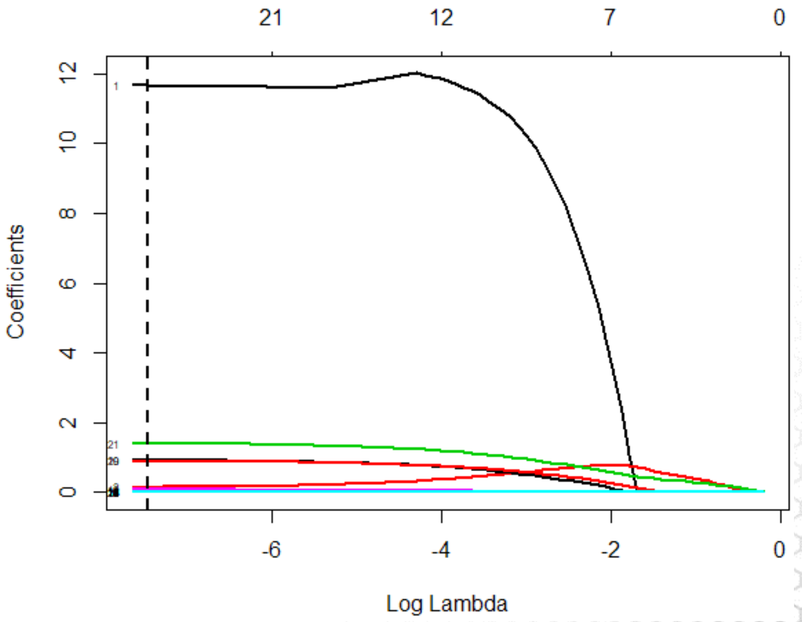
Elastic Net Regression

(Intercept)	2.288092e+00
HO	1.165709e+01
PO	1.478576e-01
WhitePop	3.688873e-03
BlackPop	4.184739e-03
HealthyPop	-1.170339e-03
ChronicPop	-5.767968e-03
Unemployment	3.568585e-04
Income	-2.361412e-07
Poverty	-2.646852e-04
Education	-1.451879e-03
Accessibility	-1.859399e-03
Availability	7.703073e-02
RankingsPCP	7.168545e-04
RankingsFood	5.944554e-03
RankingsHousing	-4.569033e-03
RankingsExercise	4.221634e-04
RankingsSocial	.
ProvDensity	5.941349e-02
AR	9.140417e-01
LA	8.996673e-01
NC	1.404530e+00
suburban	-3.213212e-04
urban	2.105330e-02

High-coefficient path corresponds to *HO* variable

RankingsSocial dummy variable is not selected

Other large-coefficient paths correspond to State dummy variables (*AR*, *LA*, *NC*)



Stepwise Regression

```
full = lm(log(EDCost.pmpm) ~ HealthyPop + ChronicPop + State + Urbanicity + HO + PO +
        BlackPop + WhitePop + Unemployment + Income + Poverty+ Education +
        Accessibility + Availability + ProvDensity +
        RankingsPCP + RankingsFood + RankingsExercise + RankingsSocial, data=dataAdult)
minimum = lm(log(EDCost.pmpm) ~ HealthyPop + ChronicPop, data=dataAdult)
# Forward Stepwise Regression
forward.model = step(minimum, scope=list(lower=minimum, upper=full), direction="forward")
summary(forward.model)
# Backward Stepwise Regression
backward.model = step(full, scope=list(lower=minimum, upper=full), direction = "backward")
summary(backward.model)
# Forward-Backward Stepwise Regression
both.min.model = step(minimum, scope=list(lower=minimum, upper=full), direction = "both")
summary(both.min.model)
```

Stepwise Regression

Observations

- Variables not selected:
  - *Unemployment*, *Income*, *Poverty*, *RankingExercise, RankingsSocial*
- *Urbanicity* was not statistically significant
- Variables selected first by forward stepwise regression, in order
  - State dummy variables (*StateAR, StateLA, StateNC*)
  - Number of inpatient claims per-member-per-month

Stepwise Regression Model

```
                    Estimate Std. Error t value Pr(>|t|)    
(Intercept)        2.0271089  0.0995378  20.365  < 2e-16 ***
HealthyPop        -0.0005092  0.0007837  -0.650 0.515917    
ChronicPop        -0.0051250  0.0020252  -2.531 0.011418 *  
StateAR            0.9324593  0.0155667  59.901  < 2e-16 ***
StateLA            0.9003846  0.0118631  75.898  < 2e-16 ***
StateNC            1.4268425  0.0157605  90.533  < 2e-16 ***
HO                12.0476486  0.7237072  16.647  < 2e-16 ***
Education         -0.0016689  0.0002312  -7.218 6.08e-13 ***
ProvDensity        0.0605923  0.0156154   3.880 0.000106 ***
RankingsPCP        0.0007885  0.0001577   5.000 5.94e-07 ***
Availability       0.0756249  0.0191618   3.947 8.03e-05 ***
Accessibility     -0.0019930  0.0007001  -2.847 0.004433 ** 
PO                 0.1232428  0.0406869   3.029 0.002466 ** 
UrbanicitySuburban -0.0017746  0.0136754  -0.130 0.896758    
UrbanicityUrban    0.0226383  0.0124409   1.820 0.068870 .  
BlackPop           0.0050790  0.0005596   9.076  < 2e-16 ***
WhitePop           0.0046371  0.0005522   8.398  < 2e-16 ***
RankingsFood       0.0158764  0.0040770   3.894 9.98e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2322 on 5001 degrees of freedom
Multiple R-squared:  0.8483,	Adjusted R-squared:  0.8478 
F-statistic:  1645 on 17 and 5001 DF,  p-value: < 2.2e-16
```

Both models explain the same amount of variance (about 84%). Prefer the smaller model.

Urbanicity is not statistically significant at $\alpha = 0.05$.

Access to primary care (*Accessibility* and *Availability*) is statistically significantly associated to ED cost.

Stepwise Regression Vs Full Models

Compare full model to selected model

```
reg.step = lm(log(EDCost.pmpm) ~ HealthyPop + ChronicPop + State +  Urbanicity + HO
      + PO + BlackPop + WhitePop + Education +  Accessibility + Availability
          + ProvDensity + RankingsPCP + RankingsFood, ,data=dataAdult)

anova(reg.step, full)
```

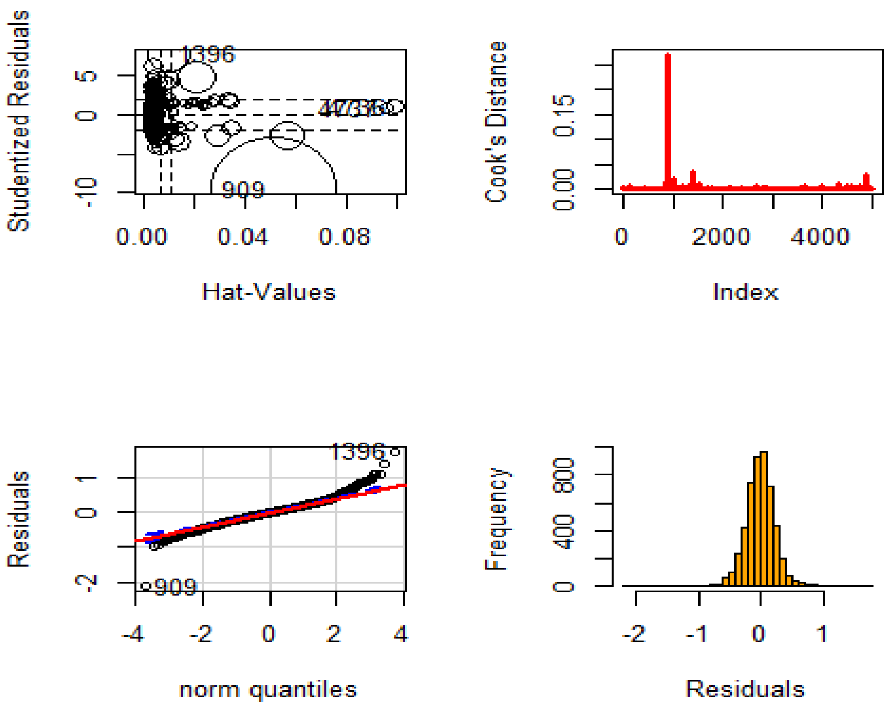
	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	5001	269.56				
2	4996	269.46	5	0.10406	0.3859	0.8588

- P-value large
  - Do not reject the null hypothesis (reduced model)
- The reduced model is plausibly as good in terms of explanatory power as the full model

Residual Analysis: Outliers & Normality

```
red.resid = rstandard(reg.step)
red.cook = cooks.distance(reg.step)

## Check outliers
influencePlot(reg.step)
plot(red.cook,type="h",lwd=3,col="red", ylab = "Cook's Distance")

## Check normality
qqPlot(red.resid, ylab="Residuals", main = "")
qqline(red.resid, col="red", lwd=2)
hist(red.resid, xlab="Residuals", main = "", nclass=30, col="orange")
```

Residual Analysis: Outliers & Normality



Outliers

Observation 909 stands out

Normality

Symmetric but with heavy tails

Removing Outlier?

Regression Output: With Outlier

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	2.0271089	0.0995378	20.365	< 2e-16
HealthyPop	-0.0005092	0.0007837	-0.650	0.515917
ChronicPop	-0.0051250	0.0020252	-2.531	0.011418
StateAR	0.9324593	0.0155667	59.901	< 2e-16
StateLA	0.9003846	0.0118631	75.898	< 2e-16
StateNC	1.4268425	0.0157605	90.533	< 2e-16
UrbanicitySuburban	-0.0017746	0.0136754	-0.130	0.896758
UrbanicityUrban	0.0226383	0.0124409	1.820	0.068870
HO	12.0476486	0.7237072	16.647	< 2e-16
PO	0.1232428	0.0406869	3.029	0.002466
BlackPop	0.0050790	0.0005596	9.076	< 2e-16
WhitePop	0.0046371	0.0005522	8.398	< 2e-16
Education	-0.0016689	0.0002312	-7.218	6.08e-13
Accessibility	-0.0019930	0.0007001	-2.847	0.004433
Availability	0.0756249	0.0191618	3.947	8.03e-05
ProvDensity	0.0605923	0.0156154	3.880	0.000106
RankingsPCP	0.0007885	0.0001577	5.000	5.94e-07
RankingsFood	0.0158764	0.0040770	3.894	9.98e-05

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '

Residual standard error: 0.2322 on 5001 degrees of freedom
Multiple R-squared:  0.8483,   Adjusted R-squared:  0.8478
```

Regression Output: Without Outlier

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	1.9356344	0.0991296	19.526	< 2e-16
HealthyPop	0.0003798	0.0007824	0.485	0.627430
ChronicPop	-0.0010849	0.0020519	-0.529	0.597031
StateAR	0.9379139	0.0154403	60.745	< 2e-16
StateLA	0.8989533	0.0117596	76.444	< 2e-16
StateNC	1.4282364	0.0156224	91.422	< 2e-16
UrbanicitySuburban	-0.0006647	0.0135555	-0.049	0.960895
UrbanicityUrban	0.0222961	0.0123314	1.808	0.070654
HO	11.5397384	0.7193214	16.043	< 2e-16
PO	0.1338608	0.0403440	3.318	0.000913
BlackPop	0.0050502	0.0005547	9.105	< 2e-16
WhitePop	0.0044178	0.0005478	8.064	9.14e-16
Education	-0.0017147	0.0002292	-7.480	8.72e-14
Accessibility	-0.0018658	0.0006940	-2.688	0.007205
Availability	0.0755848	0.0189930	3.980	7.00e-05
ProvDensity	0.0654339	0.0154862	4.225	2.43e-05
RankingsPCP	0.0007560	0.0001564	4.835	1.37e-06
RankingsFood	0.0162198	0.0040412	4.014	6.07e-05

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '

Residual standard error: 0.2301 on 5000 degrees of freedom
Multiple R-squared:  0.8504,   Adjusted R-squared:  0.8499
```

Model Interpretation: State Differences

Comparing 2011 ED Costs by Location (AL, AR, LA, and NC)

- Controlling for utilization, access, and socioeconomics
  - In AR versus AL
    - ED cost PMPM is exp(0.938) = $2.55 higher
    - ED cost per member per year is $30.65 higher
  - In LA versus AL
    - ED cost PMPM is exp(0.899) = $2.46 higher
    - ED cost per member per year is $29.49 higher
  - In NC versus AL
    - ED cost PMPM is exp(1.428) = $4.17 higher
    - ED cost per member per year is $50.04 higher

Overall Interpretation: Controlling for many potential factors contributing to ED costs, North Carolina pays significantly more while Alabama pays significantly less per member on emergency care than do Louisiana and Arkansas.

Model Interpretation: Utilization

Healthcare Utilization

- *PO*
  - Proxy of regular care utilization
  - Number of claims reimbursed for care in a physician's office
- *HO*
  - Proxy of inpatient care utilization
  - Number of claims reimbursed for hospital care

Interpretation

- An increase of 1 claim PMPM for regular care results in a 0.133 increase in log of ED cost PMPM, given all other predictors fixed
- An increase of 1 claim PMPM for inpatient care results in a 11.54 increase in log of ED cost PMPM, given all other predictors fixed

Model Interpretation: Access to Care

Access to primary care

- *Availability*
  - Proxy of wait times for appointment
  - Takes values between 0 (low wait time) and 1 (high wait time)
- *Accessibility*
  - Travel distance to primary care providers, measured in miles

Interpretation

- An increase of 0.01 or 1% in lack of availability of primary care providers results in 0.000755 unit increase in log(ED cost PMPM) given all other predictors fixed
- A reduction of 1 mile in travel distance to primary care providers results in 0.002 unit increase in log(ED cost PMPM) given all other predictors fixed
- The correlation between the two measures is 0.696. If *Availability* is discarded from the model, *Accessibility* is not statistically significant.

Georgia Tech

Summary

Regression Analysis

Regression Analysis in Practice

Nicoleta Serban, Ph.D.
Professor
Stewart School of Industrial and Systems Engineering

Emergency Department Healthcare Costs: Findings

Georgia Tech

About This Lesson

Access to Care: Intervention

Access to Primary Care

- *Availability*
  - Proxy for appointment wait times
  - Takes values between 0 (low wait times) and 1 (high wait times)

Interpretation

- An increase of 1% in lack of availability of primary care providers results in $1.00075 unit increase in ED cost PMPM, given all other predictors fixed

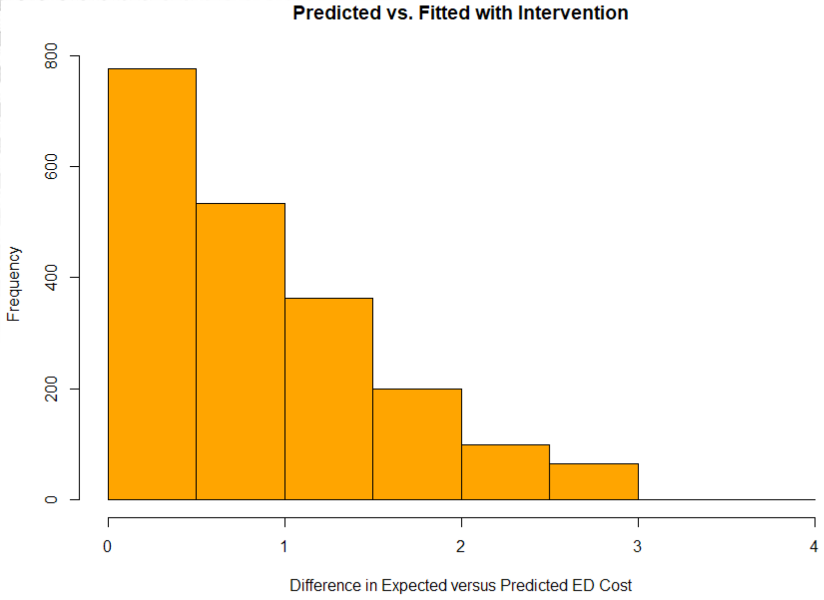
Policy Research Question

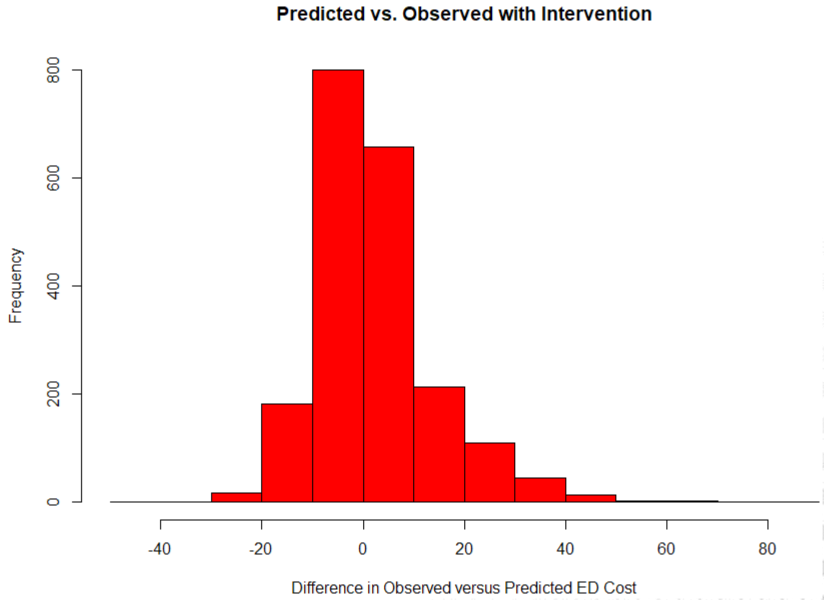
- Does improvement in availability of primary care providers reduce the cost of ED care?

Findings: Access Intervention

```
newdata=dataAdult.no.out
index = which(newdata$Availability >= 0.5)

# Improve Availability to at most 0.5 congestion experienced by all communities
newdata$Availability[index] = 0.5

# Predict by changing Availability with all other predictors fixed
EDCost.predict = predict(reg.step.no.out, newdata, interval="prediction")[,1]

# Compare predicted to fitted for those communities with intervention
EDCost.diff.fitted = exp(fitted(reg.step.no.out)) - exp(EDCost.predict)
hist(EDCost.diff.fitted[index], xlab="Difference in Expected versus Predicted ED Cost",
	main="Predicted vs. Fitted with Intervention", col="orange")

# Compare predicted to observed for those communities with intervention
EDCost.diff.observed = EDCost.pmpm[-909] - exp(EDCost.predict)
summary(EDCost.diff.observed[index])
hist(EDCost.diff.observed[index], xlab="Difference in Observed versus Predicted ED Cost",
	main="Predicted vs. Observed with Intervention", col="red")
```

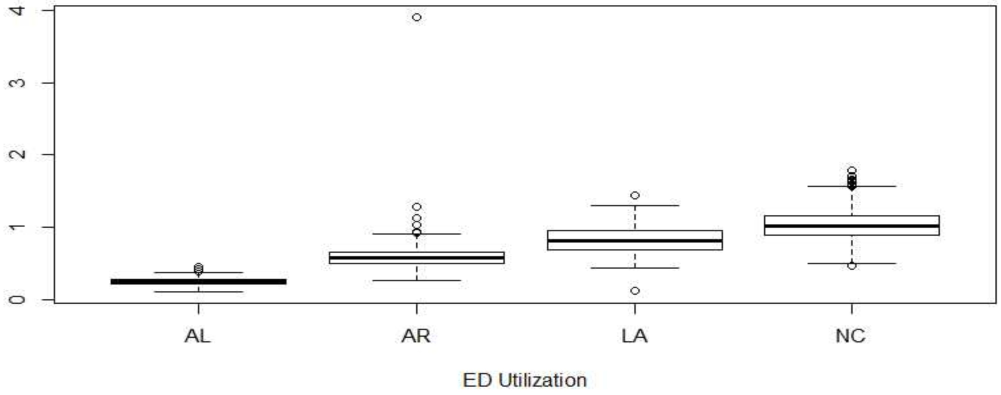
Findings: Access Intervention





Findings: State Variations

- Large variations in ED healthcare cost across the four states
  - North Carolina leads and Alabama trails in ED care cost. ***Why?***
    - Medicaid programs vary by state
      - Different health policies and reimbursements levels
    - North Carolina leads and Alabama trails also in ED utilization PMPM



The correlation between ED cost and ED utilization is 0.899

Findings: Utilization

- Utilization of physician office visits is positively associated with ED cost of care given the other predicting variables fixed in the model
  - Correlation between utilization of physician office visits and utilization of ED is high (0.54)
    - There may be communities with higher utilization of healthcare in general and thus higher ED costs

- Utilization of inpatient care (hospitalizations) is positively associated with ED cost of care given the other predicting variables fixed in the model
  - There is a very weak correlation between utilization of inpatient care and utilization of ED
  - Further investigation is needed

Georgia Tech

Findings: Other Variables

- *Education* is the only socioeconomic variable selected in the reduced model
  - Other socioeconomic variables do not add additional explanatory power given the other predicting variables in the model

- Availability of primary care providers is statistically significantly associated with ED cost of care
  - Intervening to improve availability shows a reduction in the expected ED cost of care according to the fitted model
    - Such analysis relies on causal inference

- Whether living in urban or rural communities is not statistically significantly associated to ED cost of care given other predicting variables in the model

Summary

Regression Analysis

Regression Analysis in Practice

Nicoleta Serban, Ph.D.
Professor
School of Industrial and Systems Engineering

Customer Churn Analysis in the Telecom Sector

About This Lesson

Customer Churn Analysis

Customer Churn is of great interest in industries where revenues are heavily dependent on subscriptions.

Dataset: Customer data for 7,043 telecom clients, all located in CA, USA.

Data Source: IBM Business Analytics Community

Acknowledgement: *This example was prepared with support from students in the Masters of Analytics program, including Jared Babcock, Rishi Bubna, Marta Bras, Aymee Garcia Lopez Gavilan, and Artur Bessa Cabral.*

Georgia Tech

Response & Predicting Variables

Response variables:

- *Churn Value:* 1 = the customer left the company. 0 = the customer remained with the company.

Predicting variables:

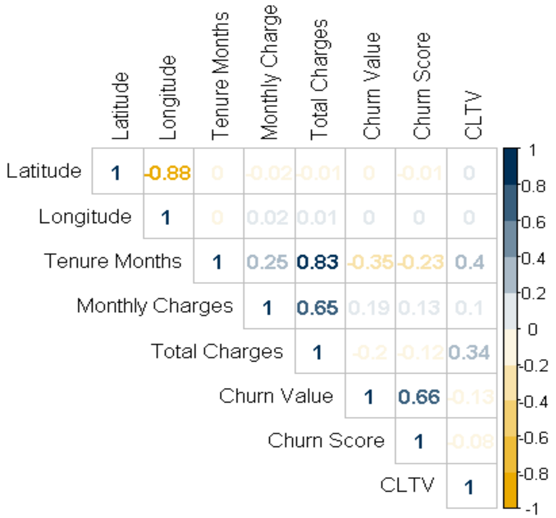
- **Demographics:** 4 variables including customer's gender (*Gender*), marital status (*Partner*) among others.
- **Location:** 7 variables including customer's primary residence ZIP Code (*Zip Code*), latitude (*Latitude*) among others.
- **Services:** 15 variables including customer's subscriptions to home phone (*Phone Services*), internet (*Internet services*), tech support (*Tech Support*) among others services.
- **Status:** 6 variables including customer's ID (*CustomerID*), reason for leaving the company (*Churn Reason*), customer's lifetime value (CLTV) among others.

Objective and Methods

- Predict which customers are likely to churn.
  - Logistic Regression
  - K Nearest Neighbors
  - Decision Tree
  - Random Forest

Exploratory Data Analysis in R

```
## Correlation among the numeric variables
# Select numerical variables
dat.num <- na.omit(dat[ , which(sapply(dat, is.numeric))])

# Create correlation matrix
corr <- cor(dat.num)

# Create correlation plot
col <- colorRampPalette(c(buzzgold,"white", gtblue))(10)
corrplot(corr, method = "number", type = "upper",
         tl.col="black", col = col)
```



There appears to be strong correlation among some of the predicting variables.

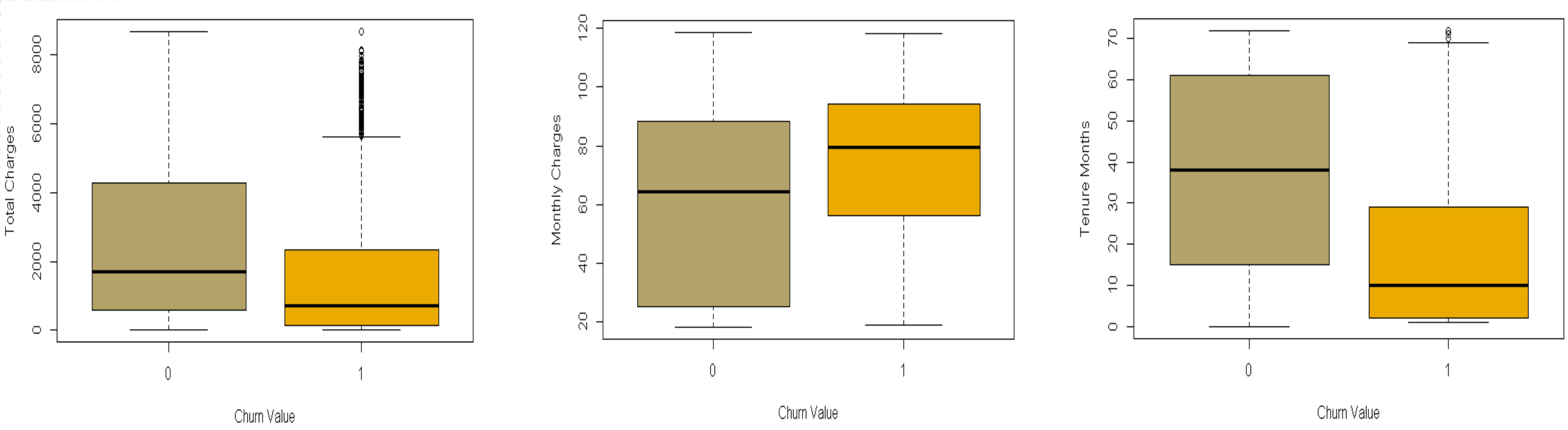
Georgia Tech

Exploratory Data Analysis in R (cont'd)

```
## Relationship between binary response and numerical variables

par(mfrow=c(2,2))
boxplot(Total.Charges ~ Churn.Value, main="", xlab="Churn Value", ylab="Total Charges",
col=c(techgold,buzzgold), data=dat)

boxplot(Monthly.Charges ~ Churn.Value, main="", xlab="Churn Value", ylab="Monthly Charges",
col=c(techgold,buzzgold), data=dat)

boxplot(Tenure.Months ~ Churn.Value, main="", xlab="Churn Value", ylab="Tenure Months",
col=c(techgold,buzzgold), data=dat)
```

Georgia Tech

Exploratory Data Analysis in R (cont’d)



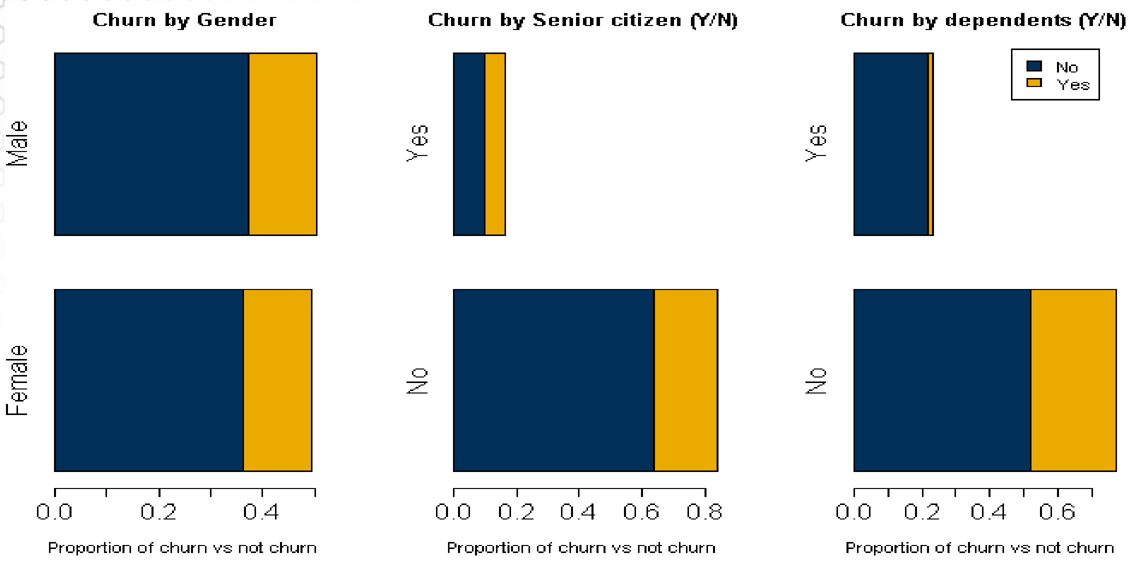
Customers that remain with the company appear to have higher total charges and tenure months but lower monthly charges than customers that have churned.

Exploratory Data Analysis in R (cont'd)

```
## Relationship between binary response and categorical variables
par(mfrow=c(1,3))

tb_obgender = xtabs(~dat$Churn.Value+ dat$Gender)
barplot(prop.table(tb_obgender),axes=T,space=0.3,  cex.axis=1.5, cex.names=1.5,
            xlab="Proportion of churn vs not churn",
            horiz=T, col=c(gtblue,buzzgold),main="Churn by Gender")

tb_citizen = xtabs(~dat$Churn.Value+ dat$Senior.Citizen)
barplot(prop.table(tb_citizen),axes=T,space=0.3, cex.axis=1.5, cex.names=1.5,
            xlab="Proportion of churn vs not churn",
            horiz=T, col=c(gtblue,buzzgold),main="Churn by Senior citizen (Y/N)")

tb_Dependents = xtabs(~dat$Churn.Value+ dat$Dependents)
barplot(prop.table(tb_Dependents),axes=T,space=0.3,cex.axis=1.5, cex.names=1.5,
            xlab="Proportion of churn vs not churn ",
           horiz=T, col=c(gtblue,buzzgold),main="Churn by dependents (Y/N)",
           legend.text = c("No", "Yes"))
```

Georgia Tech

Exploratory Data Analysis in R (cont'd)



There seems to exist significant differences in the proportions for each group in the predicting variables *Senior Citizen* and *Dependents*.

Summary

Regression Analysis

Regression Analysis in Practice

Nicoleta Serban, Ph.D.
Professor
School of Industrial and Systems Engineering

Predicting Churn Values of Customers: Regression & Variable Selection

Georgia Tech

About This Lesson

Logistic Regression

```
## Create full model
full.model <- glm(Churn.Value~ ., family = "binomial", data =
train)
summary(full.model)
## Finding insignificant variables
which(summary(full.model)$coeff[,4]>0.05)

## The overall regression seems to have explanatory power
## Model Assessment: Multicollinearity
vifs <- vif(full.model)
```

Not statistically significant in the full model:
Gender, Senior Citizen, Phone Service, Multiple Lines, Internet Service, Online Security, Online Backup, Device Protection, Tech Support, Streaming TV, Streaming Movies, Payment Method, Monthly Charges

Logistic Regression (cont'd)

Create full model
full.model <- glm(Churn.Value~ ., family = "binomial", data = train)
summary(full.model)
Finding insignificant variables
which(summary(full.model)$coeff[,4]>0.05)

The overall regression seems to have explanatory power
Model Assessment: Multicollinearity
vifs <- vif(full.model)

	GVIF	Df	GVIF^(1/(2*Df))
Gender	1.003414	1	1.001705
`Senior Citizen`	1.112401	1	1.054704
Partner	1.248636	1	1.117424
Dependents	1.098666	1	1.048173
`Tenure Months`	15.612548	1	3.951272
`Phone Service`	35.526189	1	5.960385
`Multiple Lines`	7.434935	1	2.726708
`Internet Service`	382.924211	2	4.423624
`Online Security`	5.158636	1	2.271263
`Online Backup`	6.520493	1	2.553526
`Device Protection`	6.611606	1	2.571304
`Tech Support`	5.409603	1	2.325855
`Streaming TV`	25.075402	1	5.007534
`Streaming Movies`	25.317771	1	5.031677
Contract	1.625406	2	1.129121
`Paperless Billing`	1.128532	1	1.062324
`Payment Method`	1.413278	3	1.059346
`Monthly Charges`	694.903171	1	26.361016
`Total Charges`	20.166529	1	4.490716

Variable Selection

Reduce the number of factors in the model

1. Overfitting
   - Model with large # of factors can fit too closely, cause random effects
   - It can cause bad estimates
2. Simplicity
   - Less chance of insignificant factors
   - Easier to interpret

Variable Selection (cont'd)

- Forward-Backward Stepwise Regression

```
# Create minimum model including an intercept
min.model <-  glm(Churn.Value~ 1, family = "binomial", data = train)
# Perform stepwise regression
step.model <- step(min.model, scope = list(lower = min.model, upper = full.model),
                   direction = "both", trace = FALSE)
```

- **Not selected**: Gender, Senior Citizen, Online Backup, Device Protection, Monthly Charges
- **Not statistically significant**: Payment Method by Mailed check and by Credit Card

Variable Selection (cont'd)

- LASSO Regression

```
# Set predictors and response to correct format
x.train <- model.matrix(Churn.Value ~ ., train)[,-1]
y.train <- train$Churn.Value
# Use cross validation to find optimal lambda
cv.lasso <- cv.glmnet(x.train, y.train, alpha = 1, family = "binomial")
# Train Lasso and display coefficients with optimal lambda
lasso.model <- glmnet(x.train, y.train, alpha = 1, family = "binomial")
coef(lasso.model, cv.lasso$lambda.min)
```

- Elastic Net Regression

```
# Use cross validation to find optimal lambda
cv.elnet <- cv.glmnet(x.train, y.train, alpha = 0.5, family = "binomial")
# Train Elastic Net and display coefficients with optimal lambda
elnet.model <- glmnet(x.train, y.train, alpha = 0.5, family = "binomial")
coef(elnet.model, cv.elnet$lambda.min)
```

- **Not selected for both models**: Monthly Charges

Summary

Regression Analysis

Regression Analysis in Practice

Nicoleta Serban, Ph.D.
Professor
School of Industrial and Systems Engineering

Predicting Customer Churn

About This Lesson

Prediction

```r
## Using the full model
pred.full = predict(full.model.red, newdata =
test.reduced, type = "response")
## Using the model from stepwise selection
## Variables not selected : Gender, Senior Citizen, Online
Backup and Device Protection
pred.step = predict(step.model, newdata =
test.reduced, type = "response")
## Using the model from LASSO
## Variables not selected : Online Backup and Payment
Method
pred.lasso = predict(lasso.retrained, newdata =
as.data.frame(new_test), type = "response")
## Using the model from Elastic Net (All selected)
pred.elnet = as.vector(predict(elnet.model, newx =
x.test, type = "response", s = cv.elnet$lambda.min))
```

Use classification threshold = 0.5

Predicted churn probability < 0.5 => Churn prediction = 0
Predicted churn probability > 0.5 => Churn prediction = 1

Prediction (cont'd)

Using the full model
pred.full = predict(full.model.red, newdata = test.reduced, type = "response")
Using the model from stepwise selection
Variables not selected : Gender, Senior Citizen, Online Backup and Device Protection
pred.step = predict(step.model, newdata = test.reduced, type = "response")
Using the model from LASSO
Variables not selected : Online Backup and Payment Method
pred.lasso = predict(lasso.retrained, newdata = as.data.frame(new_test), type = "response")
Using the model from Elastic Net (All selected)
pred.elnet = as.vector(predict(elnet.model, newx = x.test, type = "response", s = cv.elnet$lambda.min))

Prediction Output

Customers in Test Data	Actual Churn Value	predClass. full	predClass. step	predClas. lasso	predClass. elnet
6	0	0	0	0	0
122	1	0	0	0	0
139	0	0	0	0	0
257	0	0	0	0	0
522	0	0	0	0	0
594	1	1	1	1	1
733	0	0	0	0	0
951	0	0	0	0	0
982	1	0	0	0	0
1078	0	0	0	0	0
1091	0	0	0	0	0
1094	0	1	1	1	1
1123	0	0	0	0	0
1161	1	0	0	0	0
1249	0	0	0	0	0
1429	0	0	0	0	0

Use classification threshold = 0.5

Predicted churn probability < 0.5 => Churn prediction = 0
Predicted churn probability > 0.5 => Churn prediction = 1

Classification Accuracy

Classification Evaluation Metrics

- Accuracy:

  Proportion of response values Y_i (churn value) predicted correctly
- Sensitivity (True Positive Rate):

  Proportion of responses with Y_i = 1 (customers who left the company) predicted correctly
- Specificity (True Negative Rate):

  Proportion of responses with Y_i = 0 (customers who remained with the company) predicted correctly

	Person who left the company	**Person who remained with the company**
Classified as churned	**True Positive**	**False Positive**
Classified as not churned	**False Negative**	**True Negative**

Model Comparison via Classification Evaluation Metrics

```
## Calculate the Accuracy, the Sensitivity and the Specificity metrics to
evaluate these models at 0.5 threshold
pred_metrics = function(modelName, actualClass, predClass) {
  cat(modelName, '\n')
  conmat <- confusionMatrix(table(actualClass, predClass))
  c(conmat$overall["Accuracy"], conmat$byClass["Sensitivity"],
    conmat$byClass["Specificity"])
}
##Full model
pred_metrics("Full Model",test$Churn.Value, predClass.full)
##Stepwise selection model
pred_metrics("Stepwise Regression Model",test$Churn.Value, predClass.step)
##Lasso model
pred_metrics("Lasso Regression Model",test$Churn.Value, predClass.lasso)
##Elastic Net model
pred_metrics("Elastic Regression Model",test$Churn.Value, predClass.elnet)
```

Model Comparison via Classification Evaluation Metrics

Full Model		
Accuracy	Sensitivity	Specificity
0.8180	0.8577	0.6832

Stepwise Regression Model		
Accuracy	Sensitivity	Specificity
0.8174	0.8582	0.6807

Lasso Regression Model		
Accuracy	Sensitivity	Specificity
0.8168	0.8576	0.6799

Threshold value: 0.5

All models have very similar prediction metrics. In this case, correctly identifying positives is more important for us. Therefore, we should choose a model with higher Sensitivity.

Classification Evaluation Metrics: Different Threshold

Full Model

Accuracy	Sensitivity	Specificity
0.7742	0.9116	0.5521

Stepwise Regression Model

Accuracy	Sensitivity	Specificity
0.7776	0.9136	0.5567

Lasso Regression Model

Accuracy	Sensitivity	Specificity
0.7759	0.9111	0.5547

Threshold value: 0.3

All models have very similar prediction metrics. Sensitivity has improved while the specificity has decreased as well as the overall accuracy.

Goodness of fit

```
## Measure how well the Logistic Regression model (after variable selection through Stepwise Selection) fits on the training data
# Removing variables not selected by stepwise regression
step.predictors <-  names(coef(full.model.red)[index.step])
x.train <- as.data.frame(x.train)
train.final <- x.train[, - which(colnames(x.train) %in% step.predictors)]

# Aggregating the data
obdata.agg.n = aggregate(y.train ~ . , data = train.final, FUN = length)
obdata.agg.y = aggregate(y.train ~ . , data = train.final, FUN = sum)
dat.aggr <- cbind(obdata.agg.y, total = obdata.agg.n$y.train)

## Fitting the model
mod.aggr = glm(y.train / total ~ . , data = dat.aggr, weight = total, family = binomial)
summary(mod.aggr)
```

Georgia Tech

Goodness of fit (cont'd)

 # Find the Chi-square test statistics and the corresponding p-value to test the given null hypothesis.

res = resid(mod.aggr, type="deviance")

cbind(statistic = sum(res^2), pvalue = 1-pchisq(sum(res^2), mod.aggr$df.resid))

$$Test\ statistic = X^2 = \sum_{i=1}^{p} r_i^2 \sim \chi^2 with\ dof = n-(p+1)$$

$$P\ value = 1 - P(\chi^2_{n-(p+1)} < X^2)$$

Chi-Square Test Statistics	P-value
4180.503	1

P-value is equal to 1, so our model reasonably fits the training data.

Georgia Tech

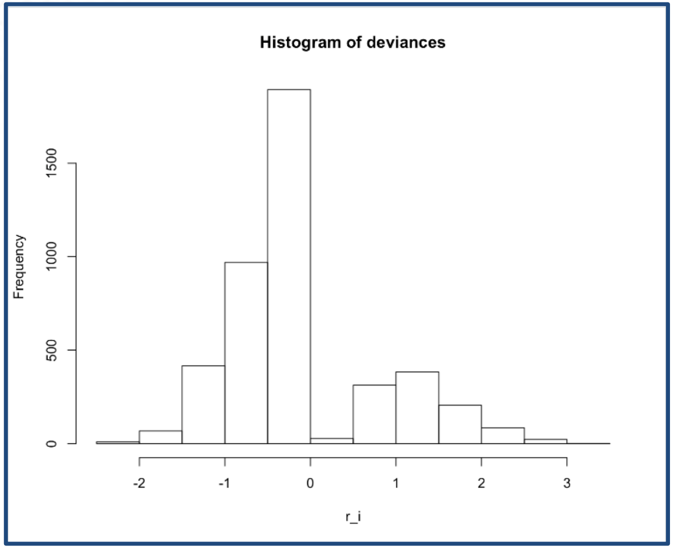
Goodness of fit (cont'd)

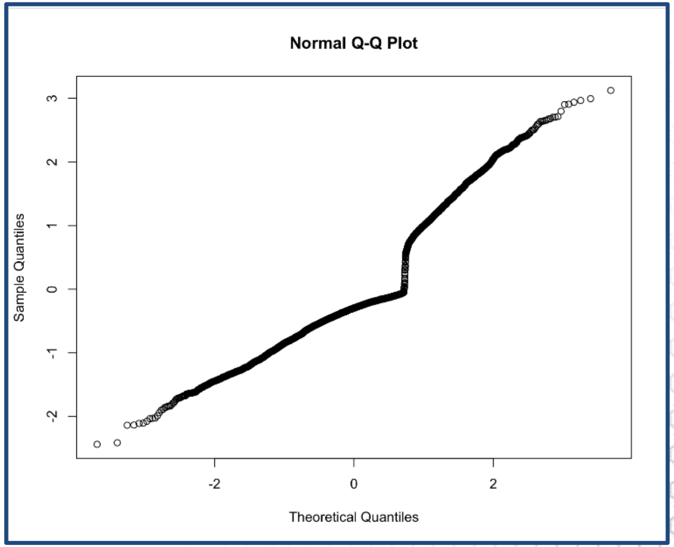
 # Checking the normality of deviance residuals assumption

hist(res, main="Histogram of deviances",breaks = 8,xlab = "r_i")

qqnorm(res)

Normality assumption seems to be violated due to bi-modality in the data.





Georgia Tech

Summary

Regression Analysis

Regression Analysis in Practice

Nicoleta Serban, Ph.D.

Professor

School of Industrial and Systems Engineering

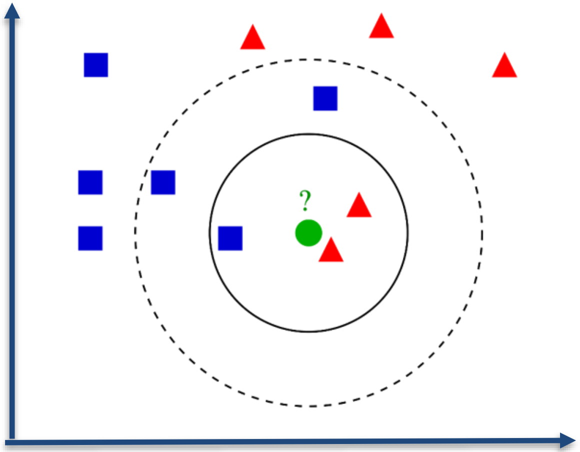
Predicting Customer Churn using Other Modeling Techniques

About This Lesson

K Nearest Neighbors (KNN): Introduction

- Classify new observations, in this case customers, according to K most similar observations.
- Class of the new observations = most found class among K nearest observations
- Supervised Learning: The labels of some observations (churn values for some customers) are known.
- Requires definition of a similarity measure (distance).

Assume that we have only two continuous features for the churn dataset. The plot represents the customer with known labels and a new customer with no information on customer's churn value



▲ : Customers who do churn

■ : Customer who do not churn

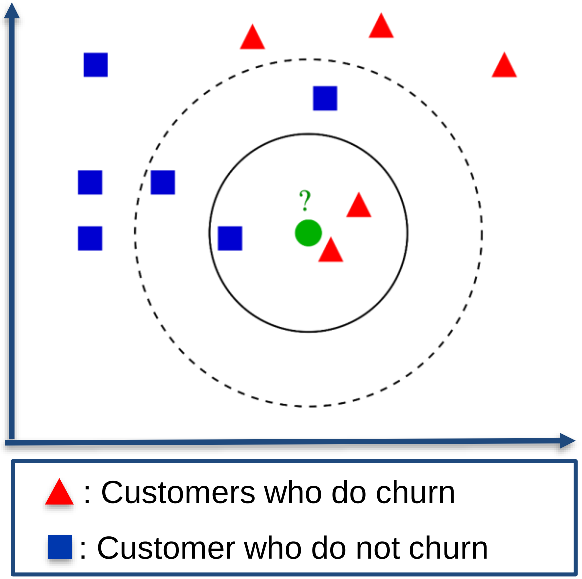
Assume the green dot represents the new customer and the contour lines represent the equal distance
from the green dot.
How do we classify the new customer if

- K = 3?
- K = 5?

Georgia Tech

K Nearest Neighbors (KNN): Introduction

- Classify new observations, in this case customers, according to K most similar observations.
- Class of the new observations = most found class among K nearest observations
- Supervised Learning: The labels of some observations (churn values for some customers) are known.
- Requires definition of a similarity measure (distance).



Defining Similarity

Assuming all features are continuous variables, let $X_{new} \in \mathbb{R}^p$ define the feature vector of new observation and X_i define the feature vectors of all available data where $i \in \{1,2,\dots,N\}$. We find the similarity between observations (customers) using

$$Similarity\ between\ the\ new\ customer\ and\ i^{th}\ customer$$
$$= \left(\sum_{k=1}^{p} (|X_{new}^k - X_i^k|)^q\right)^{1/q}$$

If

- q = 1 => Manhattan distance
- q = 2 => Euclidean distance
- q $\in$ $\mathbb{R}^+ \cup \{0\}$ => Minkowski distance

If features are categorical,

$$Similarity\ between\ the\ new\ customer\ and\ i^{th}\ customer$$
$= D_H = \sum_{i=1}^{p} D_i$ where

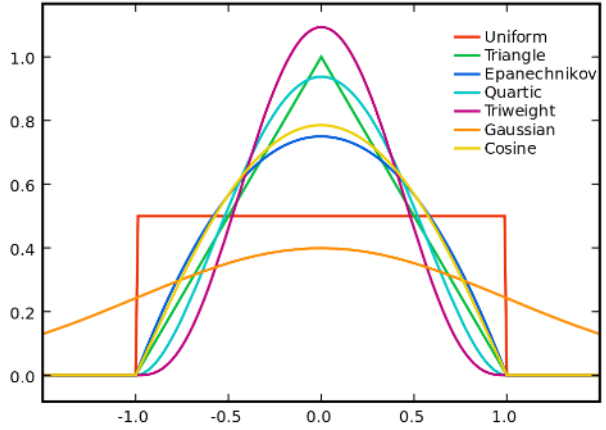
$$D_i = 0\ if\ X_{new}^k = X_i^k$$
$$D_i = 1\ if\ X_{new}^k \neq X_i^k$$

K Nearest Neighbors (KNN): Implementation

Kernel-Weighted Average Classification

What if we would like to assign more importance to the closer (more similar) observations in classifying the new observation?

Weights



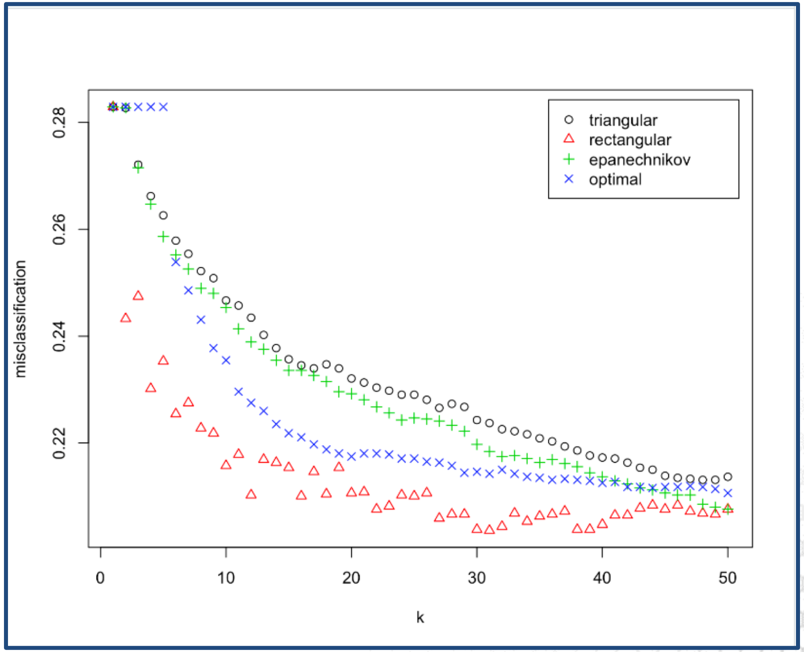
Distance

```
## Convert response to factor
y.train <- as.factor(train$`Churn Value`)
y.test <- as.factor(test$`Churn Value`)
## Dummify categorical features
dummies.train <- dummyVars(`Churn Value` ~ ., data =
train)
dummies.test <- dummyVars(`Churn Value` ~ ., data =
test)
## Create data frames containing the predictors
x.train.knn <- data.frame(predict(dummies.train, newdata =
train))
x.test.knn <- data.frame(predict(dummies.test, newdata =
test))
## Use leave-one-out cross-validation to find the
optimal value of "k"
(kknn.train <- train.kknn(y.train ~ ., x.train.knn, kmax = 50,
      kernel = c("triangular", "rectangular",
                 "epanechnikov", "optimal"),
                  scale = TRUE))
## Plot of missclassification errors vs. k for different
kernels
plot(kknn.train)
```

K Nearest Neighbors (KNN): Fitting

Leave-one-out CV

- Leave one data point out and fit a KNN model given the kernel (uniform, triangle etc.) and the value of parameter K
- Find whether the model classifies the data point correctly
- Apply the same method for all data points
- Find the misclassification rate = incorrectly classified data points / number of all data points



Optimal K = 31 and kernel = rectangular

K Nearest Neighbors (KNN): Prediction

```
# Predict the labels on the test set using Rectangular kernels and K=31
pred.knn <- predict(kknn.train, x.test.knn)

#  Calculate classification Evaluation Metrics
pred_metrics("KNN", y.test, pred.knn)
```

KNN

Accuracy	Sensitivity	Specificity
0.7957907	0.8606811	0.6158798

Chosen KNN model is more successful in identifying the people who churn compared to identifying people who do not churn.

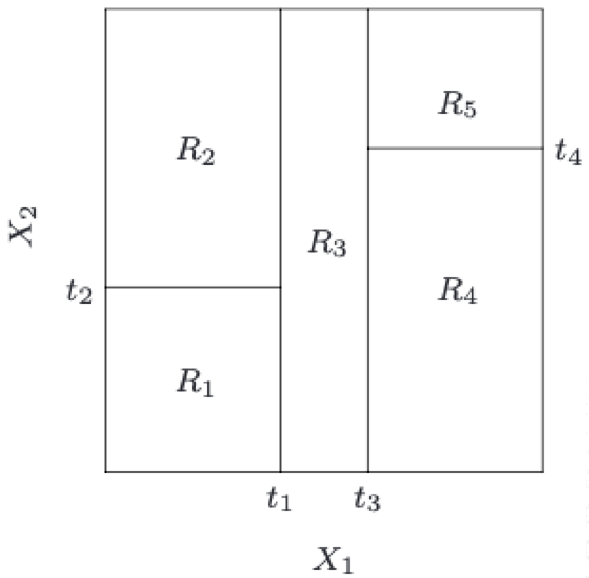
Georgia Tech

Decision Trees: Introduction

Decision Trees (DT) are non-parametric supervised learning models used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.

- Classification tree partitions the feature space into a set of rectangles
- Fit a simple model (like a constant) in each one

In our case of classification, partitioning occurs according to a specific rule and each rectangle takes the value 0 or 1 according to some other specific rule.



Partitioning in a feature space $\in \mathbb{R}^2$ with two features

Greedy Procedure for partitioning and classification under continuous features:

- Consider a splitting variable j and split point $s \in \mathbb{R}$ and define half-spaces

$$R_1(j,s) = \{X | X_J \le s\} \; \& \; R_2(j,s) = \{X | X_J > s\}$$

- Solve the problem

$$\min_{j,s}[\min_{c_1} \sum_{x_i \in R_1(j,s)} (y_i - c_1)^2 + \min_{c_2} \sum_{x_i \in R_2(j,s)} (y_i - c_2)^2]$$

$$\hat{c}_1 = majority(y_i | x_i \in R_1(j,s))$$
$$\hat{c}_2 = majority(y_i | x_i \in R_2(j,s))$$

Decision Trees: Implementation

Building model

set.seed(300)

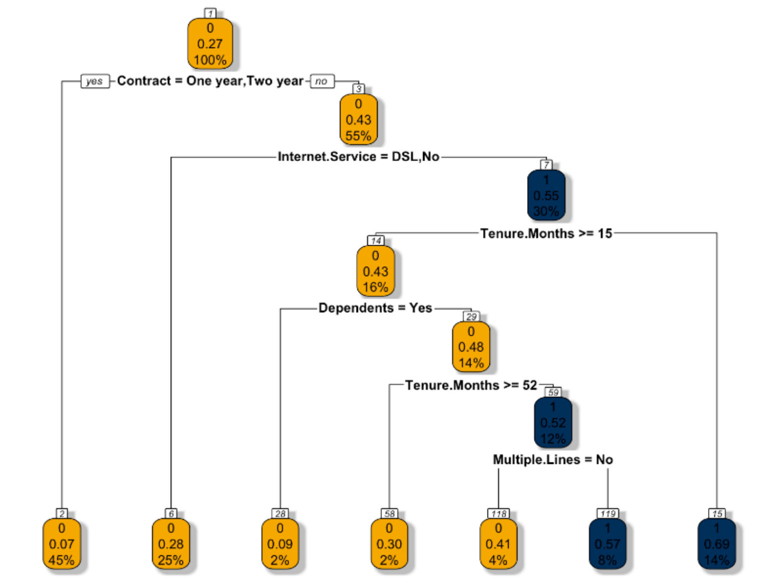
decision_tree <- rpart(Churn.Value~., data = train, method = "class")

Plotting model

rpart.plot(decision_tree, box.palette = c(buzzgold, gtblue), shadow.col = "gray", nn=TRUE)

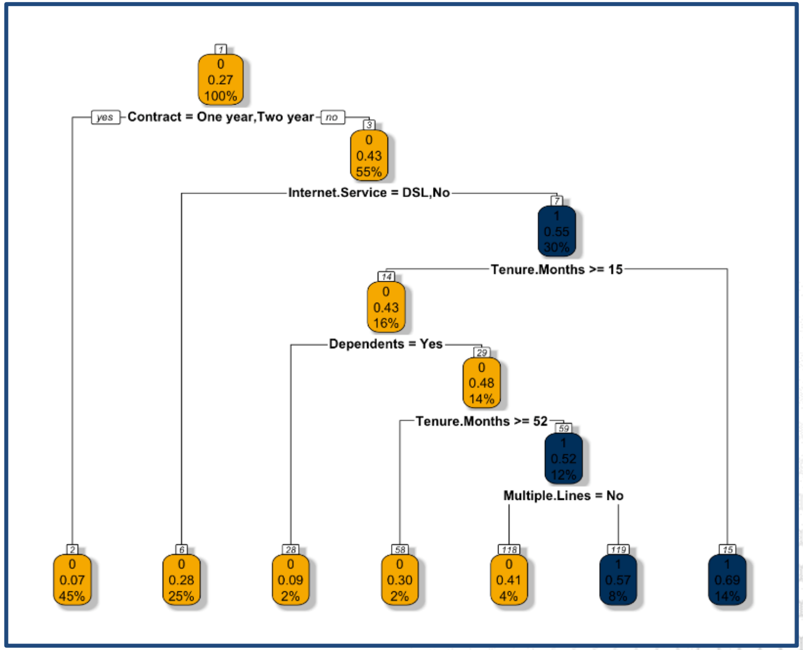
How to read the decision tree?

The first number in the node corresponds to the classification of the node (0 if not churn and 1 if churn). The second number in the node corresponds to the predicted probability of churn. The third value in the node measures the total % of customers that are included in that node.



Decision Trees: Interpretation

- The most important variable in determining churn rate is duration of contract. If the contract is 1 year, or 2 years the probability to not churn is 93%. The probability of not churning is lower if the contract is month-to-month. 45% of the total customers in the testing dataset fall in this category.
- If the customer has a month-to-month contract, has fiber optic, is in default for more than 15 months, and has dependents, then the probability of churn is only 9% ,with 2% of customers in this node.
- The higher churn occurs for month-to-month contracts, fiber optic, tenure higher than 15 months but lower than 52 months, no dependents and multiple lines. In that case, churn rate is 57%.
- Overall, the probabilities of churn are high for month-to-month contracts. The company can create incentives for customers to subscribe to longer contracts.



Georgia Tech

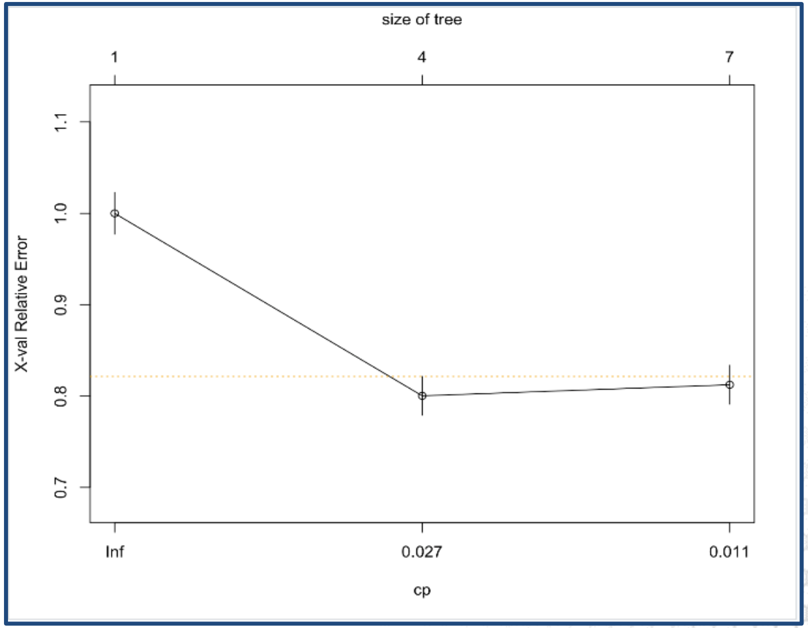
Decision Trees: Prediction

Visualize cross-validation in table format
printcp(decision_tree)

Plot the complexity parameter table
plotcp(decision_tree,minline = TRUE, lty = 3,col = buzzgold)

Predict Churn Score
predicted_churn_score <-
predict(decision_tree,test,type="class")

Create Confusion Matrix
confusionMatrix(data = as.factor(predicted_churn_score),
reference = as.factor(test$Churn.Value), positive = "1")



- Complexity Parameter finds the size of the tree that balances the size of the tree and the goodness of fit
- Note that tree size 7 minimizes the CP

Prediction\Actual	0	1
0	1220	284
1	71	183

Random Forest: Implementation

- A main issue of the tree-based method is large variance
- Trees (if grown deep enough) have low bias: can capture complicated structure but are known to be very noisy (Bias-variance tradeoff)
- Random forest: averaging

```
## Build Random Forest Model
rf <- randomForest(factor(Churn.Value)~.,data = train)

## Predict using Random Forest Model
pred_test <- predict(rf, test, type="class")

## Confusion Matrix
rf$confusion
accuracy_rf <-
(rf$confusion[1,1]+rf$confusion[2,2])/(rf$confusion[1,1]+rf$confusion[1,2]+rf$confusion[2,1]+rf$confusion[2,2])
accuracy_rf
pred_metrics("Random Forest Model",test$Churn.Value,
pred_test)
```

Random Forest		
Accuracy	Sensitivity	Specificity
0.7969283	0.8384058	0.6455026

Prediction\Actual	0	1
0	3515	357
1	684	718

In this case, the accuracy of the random forest model was just slightly better than decision tree.

Conclusion

Full Model		
Accuracy	Sensitivity	Specificity
0.8139932	0.8528551	0.6785714

Stepwise Regression Model		
Accuracy	Sensitivity	Specificity
0.8134243	0.8532649	0.6759494

Lasso Regression Model		
Accuracy	Sensitivity	Specificity
0.8156997	0.8536942	0.6828645

Elastic Net Regression Model		
Accuracy	Sensitivity	Specificity
0.8156997	0.8526623	0.6847545

Decision Tree		
Accuracy	Sensitivity	Specificity
0.798066	0.8111702	0.7204724

KNN		
Accuracy	Sensitivity	Specificity
0.7957907	0.8606811	0.6158798

Random Forest		
Accuracy	Sensitivity	Specificity
0.7969283	0.8384058	0.6455026

From the classification metrics above, we can see that both the Lasso Regression and Elastic Regression models have slightly better metrics than the other models. Therefore, those could be the chosen ones to continue to tune and work with.

Georgia Tech

Summary