Multiple Linear Regression using RStudio

Course Taught at SUAD

Dr. Tanujit Chakraborty
Ms. Madhurima Panja
@ Sorbonne
tanujitisi@gmail.com

This presentation includes…

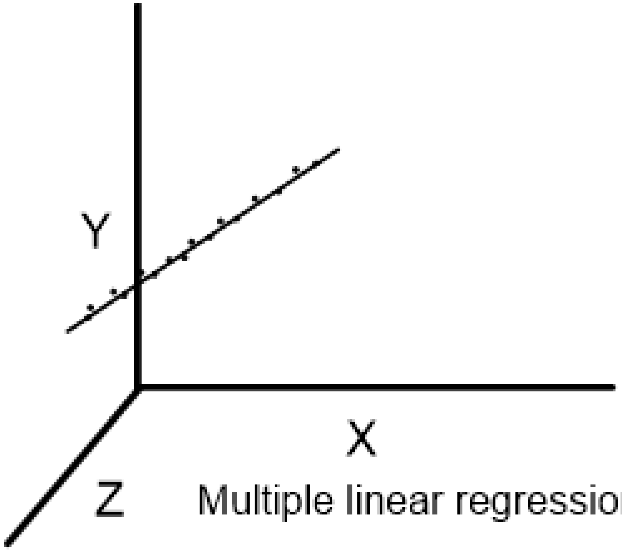
- Regression Analysis
  - Multiple Linear regression
  - Application in R
  - Multicollinearity
  - Tackling Multicollinearity
- Shrinkage Methods
  - Ridge Regression
  - LASSO

Multiple Linear Regression

Multiple Linear Regression

Definition:

- **Multiple** Regression Model**:** When more than one variable are independent variable, then the regression can be estimated as a multiple regression model
- Multiple **Linear** Regression**:** When this model is linear in coefficients, it is called multiple linear regression model



Multiple linear regression

Multiple Linear Regression

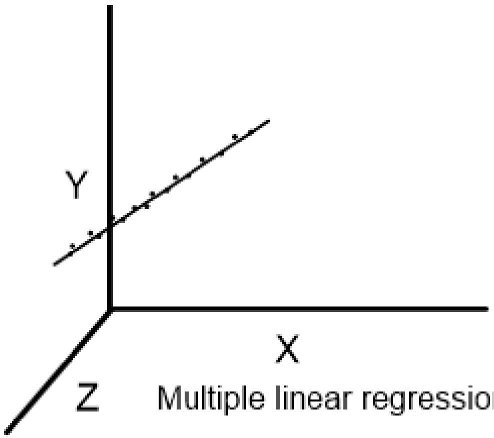
Formulation:

If *k*-independent variables $x_1, x_2, x_3 \ldots\ldots\ldots\ldots, x_k$ are associated, the multiple linear regression model is given by

$$y_i = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_k x_k + \in$$

And the estimated response is obtained as

$$\hat{y}_i = \mathrm{b}_0 + \mathrm{b}_1 x_1 + \mathrm{b}_2 x_2 + \mathrm{b}_k x_k$$



Multiple linear regression

Multiple Linear Regression

Estimating the coefficients: The data points

Let the data points given to us are

$$(x_{1i},\ x_{2i},\ \ x_{3i}, \dots \dots \dots \dots \dots \dots ,\, x_{ki}, y_i) \quad i = 1,2, \dots \dots \dots , n, \qquad n > k$$

where y_i is the observed response to the values $x_{1i},\ x_{2i},\ \ x_{3i}, \dots \dots \dots \dots \dots \dots ,\, x_{ki}$ of k independent variables $x_1,\ x_2,\ \ x_3, \dots \dots \dots \dots \dots \dots ,\, x_k$.

Multiple Linear Regression

Estimating the coefficients: The model formulation

So, the regression model in this case is given by,

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \beta_k x_{ki} + \in_i$$

and $$\hat{y}_i = \mathrm{b}_0 + \mathrm{b}_1 x_{1i} + \mathrm{b}_2 x_{2i} + \mathrm{b}_k x_{ki} + \mathrm{e}_i$$

where $\in_i$ and e_i are the random error and residual error, respectively associated with true response y_i and fitted response $\hat{y}_i$.

Multiple Linear Regression

Estimating the coefficients: Minimization of the SSE

Using the concept of **Least Square Method** to estimate $b_0, b_1, b_2, \dots, b_k$, we minimize the expression

$$\text{SSE} = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

To minimize the SSE, we need to differentiate SSE in turn with respect to $b_0, b_1, b_2, \dots, b_k$ and equate to zero.

Multiple Linear Regression

Estimating the coefficients: Minimization of the SSE

Differentiating SSE in turn with respect to $b_0, b_1, b_2, \dots, b_k$ and equating to zero, we generate the set of (k+1) normal estimation equations for multiple linear regression.

$$\mathrm{nb}_0+\mathrm{b}_1 \sum_{i=1}^{n} x_{1i} + \mathrm{b}_2 \sum_{i=1}^{n} x_{2i} + \dots + \mathrm{b}_k \sum_{i=1}^{n} x_{\mathrm{ki}} = \sum_{i=1}^{n} y_i$$

$$\mathrm{b}_0 \sum_{i=1}^{n} x_{1i} + \mathrm{b}_1 \sum_{i=1}^{n} x_{1\mathrm{i}}^2 + \mathrm{b}_2 \sum_{i=1}^{n} x_{1i}.x_{2i} + \dots + \mathrm{b}_k \sum_{i=1}^{n} x_{1i}.x_{\mathrm{ki}} = \sum_{i=1}^{n} x_i.y_i$$

$$\dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots$$

$$\dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots$$

$$\mathrm{b}_0 \sum_{i=1}^{n} x_{\mathrm{ki}} + \mathrm{b}_1 \sum_{i=1}^{n} x_{\mathrm{ki}}.x_{1i} + \mathrm{b}_2 \sum_{i=1}^{n} x_{\mathrm{ki}}.x_{2i} + \dots + \mathrm{b}_k \sum_{i=1}^{n} \mathrm{x}_{ki}^2 = \sum_{i=1}^{n} x_i.y_i$$

The system of linear equations can be solved for $b_0, b_1, \dots, b_k$ by any appropriate method for solving system of linear equations.

Mathematical Formulation of the model

A multiple linear regression model takes the form

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_{k-1} x_{k-1} + \varepsilon \quad \dots\dots\dots\dots (1)$$

- Y is the response variable
- $x_1, x_2, \dots, x_{k-1}$ are $(k-1)$ explanatory variables
- $\beta_0, \beta_1, \beta_2, \dots, \beta_{k-1}$ are regression coefficients.
- The error ε is assumed to be iid with mean 0 and variance σ^2.
- For Hypothesis testing and the setting of confidence limits, we also assume that ε is normally distributed.
- The parameters $\beta_0, \beta_1, \beta_2, \dots, \beta_{k-1}$ are sometimes called partial regression coefficients because β_j, represents the mean change in Y per unit change in x_j while all the other x variables are held constant.
- The linearity of the model (1) is defined with respect to the regression coefficients.
- We assume that the explanatory variables are known (controlled variable) and the error-free but the response variable is treated as random variable.

Linear Least Square Solution using Matrix Method

- The first objective is to estimate the k unknown parameters $\beta_0, \beta_1, \beta_2, \dots, \beta_{k-1}$.
- Suppose that we have n sets of observations of all the variables $(y_i, x_{i1}, x_{i2}, \dots, x_{ik-1}), i = 1,2, \dots, n$.
- The multiple regression model can be written as
  - $\boldsymbol{Y} = \boldsymbol{X\beta} + \boldsymbol{\varepsilon}$
- Or, equivalently,

$$\begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1k-1} \\ 1 & x_{21} & x_{22} & \cdots & x_{2k-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{nk-1} \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{k-1} \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}$$

- The vector of mean values $E[\boldsymbol{Y}]$ of $\boldsymbol{Y}$ is

$$E[\boldsymbol{Y}] = \boldsymbol{X\beta}. \text{(since E(ε) = 0)}$$

Linear Least Square Solution using Matrix Method

- The least squares solution is obtained by minimizing

$$SSResid = \sum_{i=0}^{n} e_i^2 = \boldsymbol{e}^T\boldsymbol{e} = \left(\boldsymbol{Y} - \widehat{\mathbf{Y}}\right)^T\left(\boldsymbol{Y} - \widehat{\mathbf{Y}}\right)$$

$$= \left(\boldsymbol{Y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}\right)^T\left(\boldsymbol{Y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}\right)$$

- By minimizing the above equation, we can find

$$\boldsymbol{X}^T\boldsymbol{X}\widehat{\boldsymbol{\beta}} = \boldsymbol{X}^T\mathbf{Y},$$

- $\boldsymbol{Y}$ is a $(n \times 1)$ vector of observed values.
- If the $(k \times k)$ matrix $\boldsymbol{X}^T\boldsymbol{X}$, which is symmetric, can be inverted the least squares solution to the unknown parameters is

$$\widehat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\ \boldsymbol{Y},$$

which is a $(k \times 1)$ vector of fitted parameters.

- The residual $\hat{\boldsymbol{\varepsilon}} = \mathbf{Y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}$, which constitute a $(n \times 1)$ vector of differences between the observed and estimated mean values of Y are taken as estimators of the errors

$$\boldsymbol{\varepsilon} = \mathbf{Y} - \boldsymbol{X}\boldsymbol{\beta}$$

and are used in the assessment of the model.

Properties of Least Squares Estimators

Expectations of the least squares

The least squares estimators $\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_2, \dots, \hat{\beta}_{k-1}$ are unbiased estimators of the multiple regression parameters $\beta_0, \beta_1, \beta_2, \dots, \beta_{k-1}$, i.e.,

$$E\left[\widehat{\boldsymbol{\beta}}\right] = \boldsymbol{\beta}.$$

Proof:

$$E\left[\widehat{\boldsymbol{\beta}}\right] = E[(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\ \boldsymbol{Y}] = E[(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\ (\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon})],$$

which imply,

$$E\left[\widehat{\boldsymbol{\beta}}\right] = E[\boldsymbol{\beta} + (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{\varepsilon}] = \boldsymbol{\beta}.$$

Properties of Least Squares Estimators

Covariance matrix of the least squares estimators

The covariances of the least squares estimators can be expressed as the elements of a matrix **C** as follows:

$$\boldsymbol{C} = \begin{bmatrix} Var(\hat{\beta}_0) & Cov(\hat{\beta}_0, \hat{\beta}_1) & \dots & Cov(\hat{\beta}_0, \hat{\beta}_{k-1}) \\ Cov(\hat{\beta}_0, \hat{\beta}_1) & Var(\hat{\beta}_1) & \dots & Cov(\hat{\beta}_1, \hat{\beta}_{k-1}) \\ \vdots & \vdots & & \vdots \\ Cov(\hat{\beta}_0, \hat{\beta}_{k-1}) & Cov(\hat{\beta}_1, \hat{\beta}_{k-1}) & \cdots & Var(\hat{\beta}_{k-1}) \end{bmatrix}$$

i.e. $\boldsymbol{C} = \boldsymbol{Var}(\hat{\boldsymbol{\beta}}) = \boldsymbol{E}\left[(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})^T\right]$

$$= E[\{(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\ \boldsymbol{Y} - (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{E}(\boldsymbol{Y})\}\{(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\ \boldsymbol{Y} - (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{E}(\boldsymbol{Y})\}^{\boldsymbol{T}}]$$

$$= (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T E[(\boldsymbol{Y} - E(\boldsymbol{Y}))(\boldsymbol{Y} - E(\boldsymbol{Y}))^T]\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1},$$

$$Now, \quad E[(\boldsymbol{Y} - E(\boldsymbol{Y}))(\boldsymbol{Y} - E(\boldsymbol{Y}))^T] = \begin{bmatrix} \sigma^2 & 0 & \dots & 0 \\ 0 & \sigma^2 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & \sigma^2 \end{bmatrix} = \sigma^2 \boldsymbol{I}_{n\times n}$$

Thus, $\boldsymbol{C} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\sigma^2\boldsymbol{I}_{n\times n}\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1} = \sigma^2(\boldsymbol{X}^T\boldsymbol{X})^{-1}$**.**

Hence, $$Var(\hat{\boldsymbol{\beta}}) = \sigma^2(\boldsymbol{X}^T\boldsymbol{X})^{-1}.$$

The Error Variance

- Because the error variance σ^2 is unknown, we use the residuals for estimation as given by $\boldsymbol{e} = \boldsymbol{Y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}$**.**
- The residual sum of squares is estimated as

$$SSResid = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

- In matrix notation this is represented as

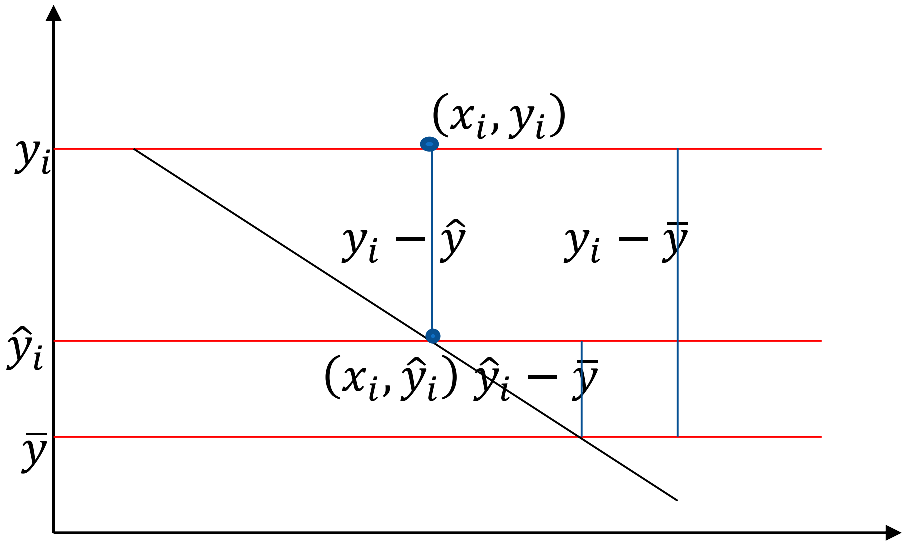
$$SSResid = \boldsymbol{e}^T\boldsymbol{e} = \left(\boldsymbol{Y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}\right)^T\left(\boldsymbol{Y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}\right) = \boldsymbol{Y}^T\boldsymbol{Y} - \widehat{\boldsymbol{\beta}}^T\boldsymbol{X}^T\boldsymbol{Y}.$$

- Because k parameters need to be estimated, an unbiased estimator of σ^2 is

$$\hat{\sigma}^2 = \frac{SSResid}{n-k} = \frac{\boldsymbol{Y}^T\boldsymbol{Y} - \widehat{\boldsymbol{\beta}}^T\boldsymbol{X}^T\boldsymbol{Y}}{n-k} = MSResid$$

- Coefficient of Multiple Determination: $R^2 = 1 - \frac{SSResid}{SSTotal}$
- $\frac{SSResid}{\sigma^2} = \frac{(n-k)\hat{\sigma}^2}{\sigma^2} \sim \chi^2_{n-k}$
- Confidence Interval of σ^2: $P\left[\frac{(n-k)\hat{\sigma}^2}{\chi^2_{n-k,\alpha/2}} \leq \sigma^2 \leq \frac{(n-k)\hat{\sigma}^2}{\chi^2_{n-k,(1-\frac{\alpha}{2})}}\right] = 1 - \alpha.$

- A set of data : (x_i, y_i).
- $\hat{Y}: predicted\ value\ of\ Y$
- $\bar{Y}: mean\ of\ Y - value$



Deviation of the i^{th} observation from the mean= Deviation of the i^{th} observation from the predicted value + Deviation of the i^{th} predicted value from the mean

$$=> \ (Y_i - \bar{Y}) = (Y_i - \hat{Y}_i) + (\hat{Y}_i - \bar{Y})$$

- Total variation in data = $\sum_{i=1}^{n}(Y_i - \bar{Y})^2$.

- The sum of the squared of the differences between each predicted y-value and $\bar{Y} = \sum_{i=1}^{n}\left(\hat{Y}_i - \bar{Y}\right)^2$
- The sum of the squared of the differences between the y-value of each ordered pair and each corresponding predicted y-value= $\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2$.

Sum of Square

- $SSResid = \boldsymbol{Y}^T\boldsymbol{Y} - \widehat{\boldsymbol{\beta}}^T\boldsymbol{X}^T\boldsymbol{Y},$ degrees of freedom of $SSResid$ is $n - k$.
- $SSTotal = \sum_{i=1}^{n} Y_i^2 - n\bar{Y}^2$, degrees of freedom of $SSTotal$ is $n - 1$.
- $SSRegression = SSReg = SSTotal - SSResid = \widehat{\boldsymbol{\beta}}^T\boldsymbol{X}^T\boldsymbol{Y} - \mathrm{n}\bar{Y}^2$, degrees of freedom of $SSReg$ is $k - 1$.
- $\frac{SSReg}{\sigma^2} \sim \chi_{k-1}^2.$

Hypothesis Testing: Model Utility Test

- To test if there is linear relationship between the response and any of the explanatory variables $x_1, x_2, \dots, x_{k-1}$.
- Hypothesis:

$$H_0 : \beta_1 = \beta_2 = \cdots = \beta_{k-1} = 0\ (slope = 0, nonlinear)$$
$$Vs.\quad H_1\ :\ \beta_j \neq 0, for\ atleast\ one\ j.$$

- ANOVA Table:

Source of Variation	D.F	SS	MS	F
Regression Residual	$k-1$ $n-k$	$SSReg$ $SSResid$	$MSReg = \frac{SSReg}{k-1}$ $MSResid = \frac{SSResid}{n-k}$	$F = \frac{MSReg}{MSResid}$ or $F = \frac{R^2/(k-1)}{(1-R^2)/(n-k)}$ $F \sim F_{k-1,n-k}$
Total	$n-1$	$SStotal$		

- Reject H_0 if $F > F_{k-1,n-k,\,\alpha}$.

Exercise 1

- The article "**Movement and Habitat Use by Lake Whitefish During Spawning in a Boreal Lake: Integrating Acoustic Telemetry and Geographic Information Systems**" (*Transactions of the American Fisheries Society* [1999]: **939–952**) included the accompanying data on 17 fish caught in 2 consecutive years.

Answer the following questions:

- Fit a multiple regression model to describe the relationship between weight and the predictors length and age.
- Interpret the value of $\beta_1 \ \ and \ \ \beta_2$.
- What is the mean weight for a fish when age is 14 year and length is 390 mm.
- Looking again Example 1, calculate the residual sum of square.
- Establish 95% confidence limit for σ^2.
- Carry out the model utility test to determine whether at least one of the predictors *length* and *age* are useful for predicting weight.

Year	Fish Nos.	Weight (g)	Length (mm)	Age (Yrs)
Year 1	1	776	410	9
	2	580	368	11
	3	539	357	15
	4	648	373	12
	5	538	361	9
	6	891	385	9
	7	673	380	10
	8	783	400	12
Year 2	9	571	407	12
	10	627	410	13
	11	727	421	12
	12	867	446	19
	13	1042	478	19
	14	804	441	18
	15	832	454	12
	16	764	440	12
	17	727	427	12

Exercise 2:

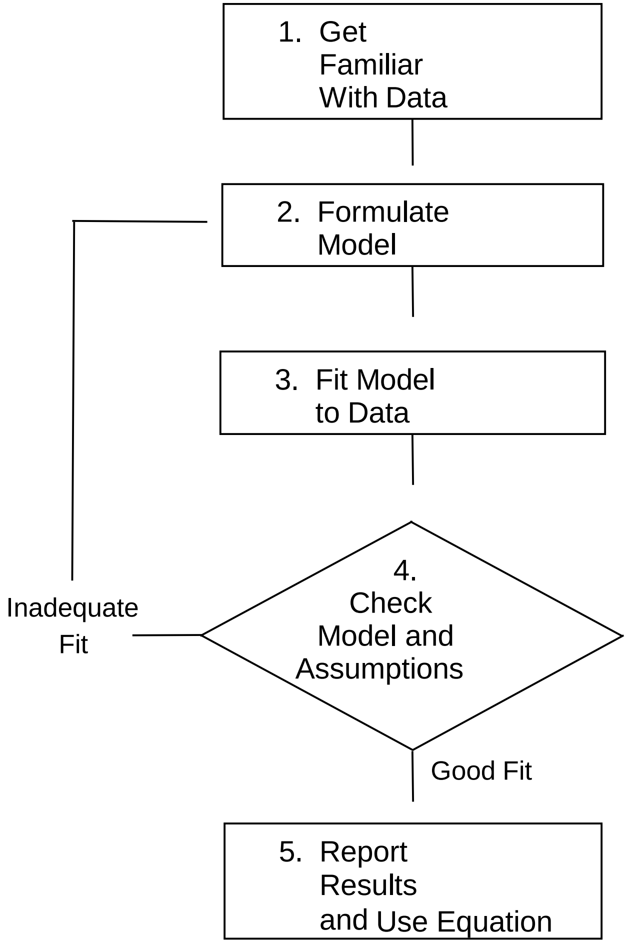
The paper **“Habitat Selection by Black Bears in an Intensively Logged Boreal Forrest” (*Canadian Journal of Zoology* [2008]: 1307–1316)** gave the accompanying data on *n* = 11 female black bears.

Age (Year)	Weight (Kg)	Home-Range Size (Km^2)
10.5	54	43.1
6.5	40	46.6
28.5	62	57.4
6.5	55	35.6
7.5	56	62.1
6.5	62	33.9
5.5	42	39.6
7.5	40	32.2
11.5	59	57.2
9.5	51	24.4
5.5	50	68.7

- Fit a multiple regression model to describe the relationship between y = home-range size and the predictors, x_1 = age and x_2 = weight.

- If appropriate, carry out a model utility test with a significance level of .05 to determine if the predictors *age* and *weight* are useful for predicting home range size.

Five Step Regression Procedure: Overview


1. Get Familiar With Data
2. Formulate Model
3. Fit Model to Data
4. Check Model and Assumptions
Inadequate Fit
Good Fit
5. Report Results and Use Equation

- Look at plots
- Look at descriptive statistics

- Linear or curvilinear?
- One X or more Xs?
- Transform?
- Discrete X, discrete Y?

- Do the regression

- Look at residuals plots
- Look at unusual observations
- Look at R-Sq
- Look at P-values for b

- Make predictions for X-values of interest

R^2 : Measure of Quality of Fit

- A quantity R^2, is called **coefficient of determination** is used to measure the proportion of variability of the fitted model.
- We have $SSE = \sum_{i=1}^{n}(y_i - \hat{y})^2$
- It signifies the **variability due to error**.
- Now, let us define the total corrected sum of squares, defined as

$$\mathrm{SST} = \sum_{i=1}^{n}(y_i - \overline{y})^2$$

- SST represents the variation in the response values. The R^2 is

$$R^2 = 1 - \frac{SSE}{SST}$$

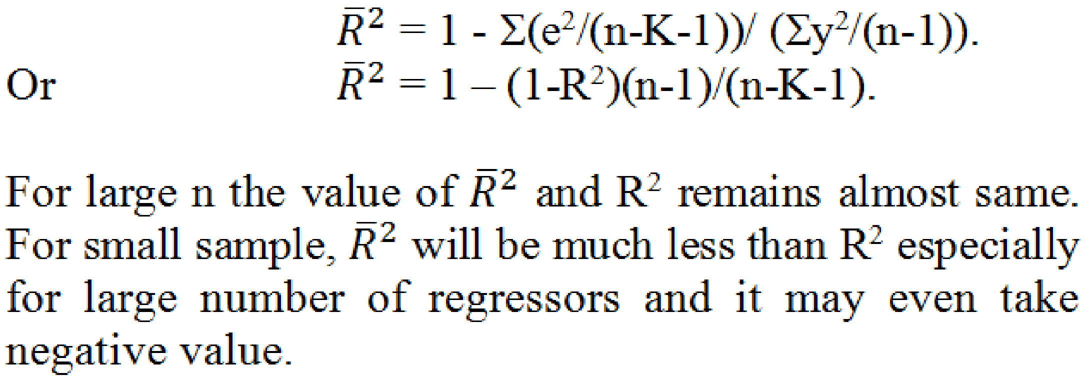
Note:

- If fit is perfect, all residuals are zero and thus $R^2 = 1.0$ (very good fit)
- If SSE is only slightly smaller than SST, then $R^2 \approx 0$ (very poor fit)
- The value of R^2 is often converted to a percentage (by multiplying by 100) and interpreted as the percentage of variation in *y* that can be explained by an approximate linear relationship between *x* and *y*.

Adjusted R^2

- The above formula for R^2 does not take into account the loss of degrees of freedom from the introduction of the additional explanatory variables in the function. The inclusion of additional explanatory variables in the function can never reduce the coefficient of multiple determination and will usually raise it.

- We introduce adjusted R^2 to compare the goodness of fit of two regression equations with different degrees of freedom. The formula for adjusted R^2 is



$$\bar{R}^2 = 1 - \Sigma(e^2/(n\text{-}K\text{-}1))/\ (\Sigma y^2/(n\text{-}1)).$$

Or $$\bar{R}^2 = 1 – (1\text{-}R^2)(n\text{-}1)/(n\text{-}K\text{-}1).$$

For large n the value of $\bar{R}^2$ and R^2 remains almost same. For small sample, $\bar{R}^2$ will be much less than R^2 especially for large number of regressors and it may even take negative value.

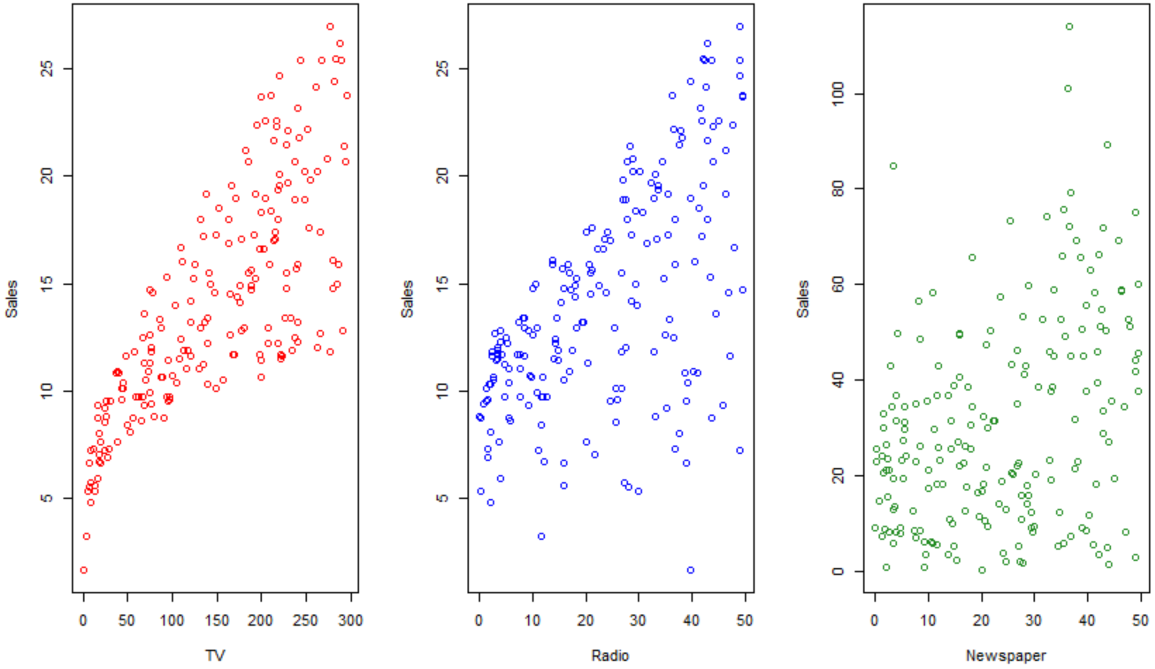
Let’s revisit the Advertisement dataset

- Suppose that we are statistical consultants hired by a client to provide advice on how to improve sales of a particular product.
- The Advertising data set consists of the sales of that product in 200 different markets, along with advertising budgets for the product in each of those markets for three different media: TV, radio, and newspaper.
- It is not possible for our client to directly increase sales of the product. On the other hand, they can control the advertising expenditure in each of the three media.
- Therefore, if we determine that there is an association between advertising and sales, then we can instruct our client to adjust advertising budgets, thereby indirectly increasing sales.
- In other words, our goal is to develop an accurate model that can be used to predict sales on the basis of the three media budgets.
- In this example, the variables TV, radio and newspaper are the feature vectors, also going by the names of predictors, covariates, independent variables, or just variables.
- The output variable, sales is the label, or the dependent variable, also called the response variable.

Visualizing the data

```
dat <- read.csv("Advertising.csv", header = TRUE)

par(mfrow=c(1,3)) # this creates three plots in a single row
plot(dat$TV, dat$sales, xlab = "TV", ylab = "Sales", col = "red")
plot(dat$radio, dat$sales, xlab="Radio", ylab="Sales", col="blue")
plot(dat$radio, dat$newspaper, xlab="Newspaper",  ylab="Sales", col="forestgreen")
```



Fitting Multiple Linear Regression Model

```
dat <- as_tibble(dat) %>% select(-X) # X represents serial number hence can be removed
# Split the data into training and testing set
set.seed(8885)
dat.split <- resample_partition(dat, c(test = 0.3, train = 0.7))
# Time for MLR model fitting
mod2 <- lm(sales ~ TV + radio + newspaper, data = train)
summary(mod2)
```

```
## Call:
## lm(formula = sales ~ TV + radio + newspaper, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.7836 -0.9319  0.2914  1.2612  2.5435
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2.7523530  0.3932178    7.00 1.04e-10 ***
## TV          0.0454035  0.0016565   27.41  < 2e-16 ***
## radio       0.1918578  0.0107023   17.93  < 2e-16 ***
## newspaper   0.0002214  0.0073347    0.03    0.976
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.73 on 137 degrees of freedom
## Multiple R-squared:  0.8911, Adjusted R-squared:  0.8888
## F-statistic: 373.8 on 3 and 137 DF,  p-value: < 2.2e-16
```

Fitting Multiple Linear Regression Model

```
# Tidy summary (available in 'broom' package)
library(broom)
tidy(mod2)

## A tibble: 4 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept) 2.75      0.393      7.00   1.04e-10
## 2 TV          0.0454   0.00166   27.4    1.81e-57
## 3 radio        0.192    0.0107    17.9    9.57e-38
## 4 newspaper  0.000221  0.00733   0.0302 9.76e- 1
```

Qn. Interpret the co-efficients. Which variables are significant?

Getting 95% confidence intervals

```
confint(mod2)
##                   2.5 %     97.5 %
## (Intercept)  1.97479189 3.52991419
## TV          0.04212795 0.04867914
## radio        0.17069469 0.21302091
## newspaper   -0.01428241 0.01472512
```

How do you interpret the coefficients

Checking Model accuracy

- We now explore goodness-of-fit of the model on the data.
- Quantitative assessment of the same may be achieved through the following measures:
  - Mean Squared Error – estimate of error variance
  - R^2 - proportion of variability explained by the predictor variable
  - F-statistic

```
glance(mod2, train)
## A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic  p.value    df logLik   AIC   BIC
##       <dbl>         <dbl> <dbl>     <dbl>    <dbl> <dbl>  <dbl> <dbl> <dbl>
## 1     0.891         0.889  1.73      374. 9.40e-66     3  -275.  561.  575.
## # ... with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

The multiple R2 came out to be 0.891142. What does it signify?

How do you interpret the coefficients

Checking Model accuracy

- We now explore goodness-of-fit of the model on the data.
- Quantitative assessment of the same may be achieved through the following measures:
  - Mean Squared Error – estimate of error variance
  - R^2 - proportion of variability explained by the predictor variable
  - F-statistic

```
glance(mod2, train)
## A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic  p.value    df logLik   AIC   BIC
##       <dbl>         <dbl> <dbl>     <dbl>    <dbl> <dbl>  <dbl> <dbl> <dbl>
## 1     0.891         0.889  1.73      374. 9.40e-66     3  -275.  561.  575.
## # ... with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

The multiple R2 came out to be 0.891142. What does it signify?

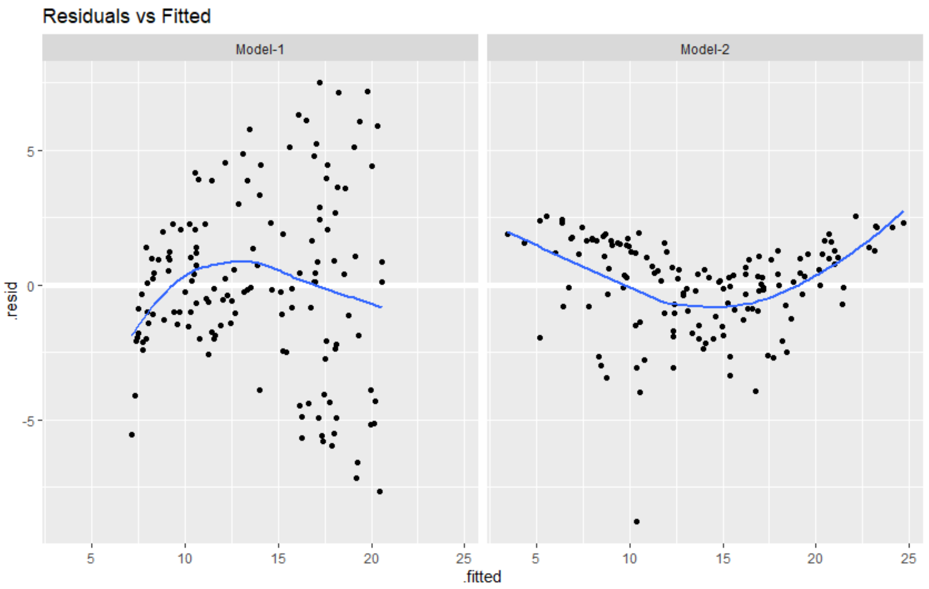
Ans: 89.11 percent of the variation in sales can be explained by the three predictors TV, radio and newspaper.

Assessment of Fitting

```
mod1_results <- augment(mod1, train) %>%
  mutate(Model = "Model-1") # SLR

mod2_results <- augment(mod2, train) %>%
  mutate(Model = "Model-2") %>%
  rbind(mod1_results) #MLR

ggplot(mod2_results, aes(.fitted, .resid)) +
  geom_ref_line(h = 0) +
  geom_point() +
  geom_smooth(se = FALSE) +
  facet_wrap(~ Model) +
  ggtitle("Residuals vs Fitted")
```



Exercise: Run the other relevant residual analysis and model diagnostics to check for normality assumptions, homoscedasticity, presence of infulential points, outliers and leverage points.

Predictions

```
test %>% add_predictions(mod2)
```

```
## # A tibble: 59 x 5
##       TV radio newspaper sales  pred
##    <dbl> <dbl>     <dbl> <dbl> <dbl>
##  1   8.7  48.9      75     7.2 12.5
##  2   8.6   2.1       1     4.8  3.55
##  3 204.   32.9      46    19   18.3
##  4  95.7   1.4       7.4   9.5  7.37
##  5 202.   22.3      31.6  16.6 16.2
##  6 207.    8.4      26.4  12.9 13.8
##  7 240.   41.5      18.5  23.2 21.6
##  8  66.9  11.7      36.8   9.7  8.04
##  9 100.    9.6       3.6  10.7  9.15
## 10 216.   41.7      39.6  22.6 20.6
## # ... with 49 more rows
```

```
MSE <- mse(mod2, train)
pred.MSE <- mse(mod2, test)
```

```
c(Prediction_MSE = pred.MSE, Train_MSE = MSE)
## Prediction_MSE      Train_MSE
##       2.556324       2.908014
```

Predictions

```
# Using base R approach (gives confidence and prediction interval of fit)
test.predict <- predict(mod2, test, interval = "prediction")
as_tibble(test.predict)

## A tibble: 59 x 3
##      fit    lwr   upr
##    <dbl>  <dbl> <dbl>
##  1 12.5   9.01   16.1
##  2  3.55  0.0468  7.05
##  3 18.3  14.9    21.8
##  4  7.37  3.90   10.8
##  5 16.2  12.8    19.7
##  6 13.8  10.3    17.2
##  7 21.6  18.1    25.1
##  8  8.04  4.59   11.5
##  9  9.15  5.69   12.6
## 10 20.6  17.1    24.0
## # ... with 49 more rows
```

Interaction Effects

In the previous example on Advertising data, we had assumed that the predictor variables were independent.

However, it might be plausible that putting an ad in the TV increases its effectiveness on the ad placed in radio.

This means that an increase in TV ad budget should impact the coefficient in radio budget as well.

We can incorporate this in our model through interaction terms.

The new model with interaction term is then given by

$$sales = \beta_0 + \beta_1 * TV + \beta_2 * radio + \beta_3 * (TV \; * radio) + \beta_4 * newspaper + \epsilon$$

```
mod3 <- lm(sales ~ TV + radio + TV:radio + newspaper, data = train)

#Aliter:
mod3 <- lm(sales ~ TV*radio + newspaper, data = train)

summary(mod3)
```

Interaction Effects

```
## Call:
## lm(formula = sales ~ TV * radio + newspaper, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.9774 -0.5146  0.1382  0.6518  1.5625
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 6.646e+00  3.321e-01  20.012   <2e-16 ***
## TV          1.869e-02  1.908e-03   9.800   <2e-16 ***
## radio       2.118e-02  1.222e-02   1.733   0.0854 .
## newspaper   5.540e-03  4.304e-03   1.287   0.2002
## TV:radio    1.118e-03  6.878e-05  16.256   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.012 on 136 degrees of freedom
## Multiple R-squared:  0.963,  Adjusted R-squared:  0.9619
## F-statistic: 885.2 on 4 and 136 DF,  p-value: < 2.2e-16
```

tidy(mod3)

```
## # A tibble: 5 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)  6.65     0.332         20.0  2.34e-42
## 2 TV           0.0187   0.00191        9.80 1.75e-17
## 3 radio        0.0212   0.0122         1.73 8.54e- 2
## 4 newspaper    0.00554  0.00430        1.29 2.00e- 1
## 5 TV:radio     0.00112  0.0000688     16.3  1.11e-33
```

1. Keeping all other variables fixed, what is the impact on sales for 1 unit increase in TV budget?

2. In radio budget?

3. How do you interpret the coefficient associated with the interaction term?

MULTICOLLINEARITY

- The use and interpretation of a multiple linear regression model often depend explicitly or implicitly on the estimates of the individual regression coefficients.

- If there is no linear relationship between the predictor variables, they are said to be **independent**, or **orthogonal**.

- Sometimes, the lack of orthogonality is not serious; however, in some situations the regressors are nearly linearly related, and in such cases the inferences based on the regression models can be misleading or errorneous.

- When there are near-linear dependence among the regressors, the problem of multicollinearity is said to exist.

MULTICOLLINEARITY

Sources of multicollinearity

- There are four primary sources of multicollinearity:
- Data collection method employed
- Constraints on the model or in the population
- Model specification
- An over defined model (large p small n problem)

Effects of multicollinearity

- Let us consider a simulated example where there is near-linear dependency among some regressors.
- First let us explore the case of exact linear relationship among some of the predictors.

MULTICOLLINEARITY

```
X1 <- runif(100)
X2 <- rnorm(100,10,10)
X3 <- rnorm(100,-20,20)
X4 <- X2 - X3 # exact linear relationship

Y <- 1 + X1 - 0.5*X2 + 0.5*X3 + rnorm(100,0,4)

dat <- as_tibble(data.frame(Y = Y, X1 = X1, X2 = X2, X3 = X3, X4 = X4))
dat
## A tibble: 100 x 5
##          Y    X1     X2     X3    X4
##      <dbl> <dbl>  <dbl>  <dbl> <dbl>
##  1 -33.0   0.874  16.5  -44.8  61.3
##  2 -11.1   0.588  -1.66 -19.0  17.3
##  3  -6.46  0.733   9.38 -12.1  21.4
##  4   9.47  0.245  -2.06  -6.25  4.19
##  5   2.26  0.628   2.85   5.96 -3.12
##  6   4.18  0.706  -5.87   1.22 -7.08
##  7  -3.08  0.894   3.62 -11.2  14.8
##  8 -22.7   0.521   1.31 -41.6  42.9
##  9   0.305 0.417 -11.5  -24.3  12.8
## 10  -5.74  0.941  -1.23 -28.9  27.6
## # ... with 90 more rows
```

MULTICOLLINEARITY

```
cor(dat)
##            Y          X1          X2          X3          X4
## Y   1.0000000  0.10038813 -0.37729361  0.85629238 -0.94944070
## X1  0.1003881  1.00000000 -0.06453975  0.07120208 -0.09202063
## X2 -0.3772936 -0.06453975  1.00000000  0.04041295  0.35636754
## X3  0.8562924  0.07120208  0.04041295  1.00000000 -0.91918069
## X4 -0.9494407 -0.09202063  0.35636754 -0.91918069  1.00000000
```

```
simu.model <- lm(Y ~ X1 + X2 + X3 + X4)

summary(simu.model)
## Call:
## lm(formula = Y ~ X1 + X2 + X3 + X4)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.2480 -2.9285 -0.5828  2.3633 12.7320
##
## Coefficients: (1 not defined because of singularities)
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.12643    0.99067   2.146   0.0344 *
## X1           0.50692    1.37973   0.367   0.7141
## X2          -0.56893    0.04398 -12.935   <2e-16 ***
## X3           0.50789    0.01855  27.382   <2e-16 ***
## X4                NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.891 on 96 degrees of freedom
## Multiple R-squared:  0.9033, Adjusted R-squared:  0.9003
## F-statistic:   299 on 3 and 96 DF,  p-value: < 2.2e-16
```

MULTICOLLINEARITY

Let us re-run this example with near-linear relationship among some of the predictors.

```
set.seed(100)
X1 <- runif(100)
X2 <- rnorm(100,10,10)
X3 <- rnorm(100,-20,20)
X4 <- X2 - X3 + runif(100) # near-linear relationship
Y <- 1 + X1 - 0.5*X2 + 0.5*X3 + rnorm(100,0,4)

dat <- as_tibble(data.frame(Y = Y, X1 = X1, X2 = X2, X3 = X3, X4 = X4))
cor(dat)
simu.model <- lm(Y ~ X1 + X2 + X3 + X4)
summary(simu.model)
```

```
## Call:
## lm(formula = Y ~ X1 + X2 + X3 + X4)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.0615  -2.3974  -0.0132   2.9371  13.9905
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.6960     1.3190   0.528    0.599
## X1            1.6438     1.6948   0.970    0.335
## X2           -0.4584     1.5809  -0.290    0.772
## X3            0.3568     1.5791   0.226    0.822
## X4           -0.1213     1.5794  -0.077    0.939
##
## Residual standard error: 4.33 on 95 degrees of freedom
## Multiple R-squared:  0.8926, Adjusted R-squared:  0.888
## F-statistic: 197.3 on 4 and 95 DF,  p-value: < 2.2e-16
```

MULTICOLLINEARITY

Notice the fact that in case of exact linear dependence, the coefficients of the variabe X_4 could not be evaluated. This is owing to the fact that the columns of the matrix $X^T X$ are linearly dependent, and hence computation of the inverse matrix was not possible.

In the second case, note that the overall model F-statistic implies significance (very low p-value), but none of the predictors are significant.

Multiple Linear Regression : Dealing with multi-collinearity

- Many predictor variables or independent variables ‘X_1, X_2, ….X_k’ (e.g.: gender, height) and a response variable or dependent variable ‘Y’ (e.g.: weight).

The regression equation is

$$\hat{Y} = a + b_1 X_1 + b_2 X_2 + \dots + b_k X_k$$

$$\text{where, } \hat{Y} = \text{Predicted value of } Y$$

$$a = \text{Intercept (the predicted value of } Y \text{ when all } X_i = 0)$$

$$b_j = \text{Slope of the line (the amount of difference in } Y \text{ associated with a 1 - unit difference in } X_j): j = 1, 2, \dots, k$$

- One of the assumption of model accuracy is that X’s are not correlated. But this may not be true always. The multi-collinearity can be checked by Variance Inflation Factor (VIF).

Variance Inflation Factor (VIF)

- The variance inflation factor (VIF) is used to detect whether one predictor has a strong linear association with the remaining predictors (the presence of multi-collinearity among the predictors).
- VIF measures how much the variance of an estimated regression coefficient increases if your predictors are correlated (multi-collinear).
- VIF = 1 indicates no relation; VIF > 1, otherwise. The largest VIF among all predictors is often used as an indicator of severe multi-collinearity.
- Montgomery and Peck suggest that when VIF is greater than 5-10, then the regression coefficients are poorly estimated.
- If VIF is greater than 5, we should consider the options to break up the multi-collinearity: collecting additional data, deleting predictors, using different predictors, or an alternative to least square regression.

Variance Inflation Factor (VIF)

```
library(car)

vif(simu.model)

##          X1          X2          X3          X4
##    1.041343 1340.110776 5145.410910 7705.062356
```

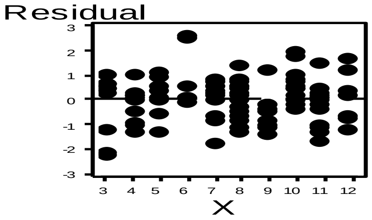
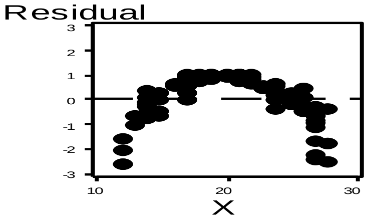
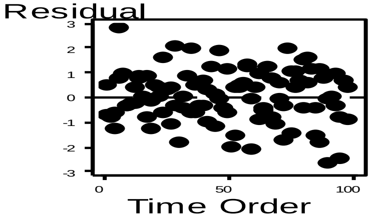
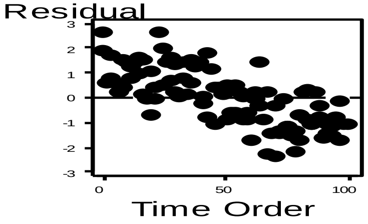
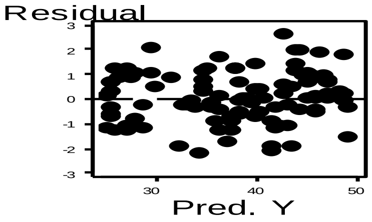
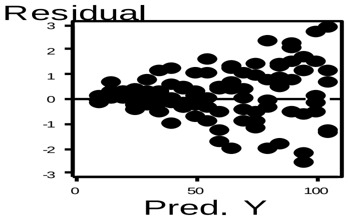
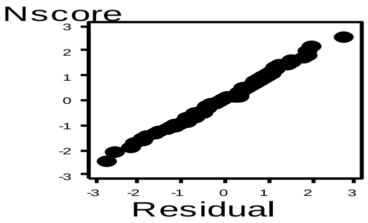
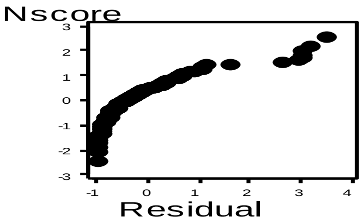
The VIF values indicates high multicollinearity

Other ways for Checking Multi-collinearity

- Matrix plot : A matrix plot is a two-dimensional matrix of individual plots. Matrix plots are good for, among other things, seeing the two-variable relationships among a number of variables all at once.
- Correlation Coefficient :
  - A measure of the relationship between variables.
  - The most commonly used coefficient is Pearson Product-Moment Correlation Coefficient (measure of linear relationship denoted by ‘r’).
  - ‘r’ lies between -1 and +1. r = 0 means no correlation.
  - If one variable tends to increase as the other decreases, the correlation coefficient ‘r’ is negative. Conversely, if the two variables tend to increase together the correlation coefficient ‘r’ is positive.
  - For a two-tailed test of the correlation:

  $H_0: r = 0$ versus $H_1: r \neq 0$ where ‘r’ is the correlation between a pair of variables.

Checking Assumptions About Residuals

Residuals Plot	Good	Bad	Meaning / Actions
1. Residuals vs Each X *Used to check that the residuals are not related to the Xs*			The relationship between X & Y is not a straight line, but a curve. Try a transformation on X, Y, or both. Or use X^2 in a multiple regression.
2. Time Plot of Residuals *Used to check for stability over time*			*Any* pattern visible over time means another factor, related to time, influences Y. Try to discover it and include it in a multiple regression.
3. Residuals vs Predicted Y (Fits) *Used to check that they are constant over the range of Ys*			This fan shape means the variation increases as Y gets larger (it's not constant). Try a square root, log, or inverse transformation on Y.
4. Normal Probability Plot of Residuals *Used to check that residuals are Normal*			The residuals are not Normal. Try a transformation on X or Y or both.

MULTIPLE LINEAR REGRESSION USING RStudio

Another Example

REGRESSION ANALYSIS

Multiple Linear Regression

To model output variable y in terms of two or more variables.

General Form:

$$y = a + b_1x_1 + b_2x_2 + - - - + b_kx_k + \varepsilon$$

Two variable case:

$$y = a + b_1x_1 + b_2x_2 + \varepsilon$$

Where

a: intercept (the predicted value of y when all x's are zero)

b_j: slope (the amount change in y for unit change in x_j keeping all other x's constant, j = 1,2,---,k)

REGRESSION ANALYSIS

Exercise : The effect of temperature and reaction time affects the X.yield. The data collected in given in the Mult-Reg_Yield file. Develop a model for X.yield in terms of temperature and time?

Step 1: Reading the data and variables

```
> mydata = read.csv(‘Mult_Reg_Yield.csv',header = T,sep = ",")
> mydata[,-1]  # Removing 1st  column
> attach(mydata)
```

REGRESSION ANALYSIS

Exercise : The effect of temperature and reaction time affects the X.yield. The data collected in given in the Mult-Reg_Yield file. Develop a model for X.yield in terms of temperature and time?

Step 2: Correlation Matrix

```
> cor(mydata)
```

	Time	Temperature	X.Yield
Time	1	-0.00756	0.89671
Temperature	-0.00756	1	-0.05457
X.Yield	0.89671	-0.05457	1

Correlation between xs & y should be high

Correlation between xs should be low

REGRESSION ANALYSIS

Step 3: Regression Output – Identify the model

```
> model = lm(X.Yield ~ Temperature + Time)
> summary(model)
```

Attribute	Coefficient	Std. Error	t Statistic	p value
Time	0.9061	0.12337	7.344	0.0000
Temperature	-0.0642	0.16391	-0.392	0.702
Intercept	-67.8844	40.58652	-1.67	0.118

Interpretation: Only time is related to % yield as p value < 0.05

REGRESSION ANALYSIS

Regression Output

Statistic	Value	Criteria
Adjusted R Square	0.7766	≥ 0.6

ANOVA

```
> anova(model)
```

Source	SS	df	MS	F	p value
Time	6777.8	1	6777.8	53.9872	0.000
Temp	19.3	1	19.3	0.1534	0.702
Residual	1632.1	13	125.5		

Criteria: P value < 0.05

REGRESSION ANALYSIS

Modified Regression Output – Identify the model

```
> model_m = lm(X.Yield ~ Time)
> summary(model_m)
```

Attribute	Coefficient	Std. Error	t Statistic	p value
Time	0.9065	0.1196	7.580	0.0000
Intercept	-81.6205	19.7906	-4.124	0.00103

Model: % Yield= 0.9065 x Time - 81.621

REGRESSION ANALYSIS

Regression Output

Statistic	Value	Criteria
Adjusted R Square	0.7901	≥ 0.6

ANOVA

```
> anova(model_m)
```

Source	SS	df	MS	F	p value
Time	6777.8	1	6777.8	57.462	0.000
Residual	1651.3	14	118.0		

Criteria: P value < 0.05

REGRESSION ANALYSIS

Step 4: Residual Analysis

```
>pred = fitted(model_m)
>Res = residuals(model_m)
>write.csv(pred,"C:/Users/Downloads/data_and_code/Data and
Code/Pred_m.csv")
>write.csv(Res,"C:/Users/Downloads/data_and_code/Data and
Code/Res_m.csv")
```

Standardizing Residuals using Scale function

```
>Std_Res = scale(Res, center = TRUE, scale = TRUE)
>write.csv(Std_Res,"C:/Users/Downloads/data_and_code/Data and
Code/Std_Res_m.csv")
```

The "center" parameter (when set to TRUE) is responsible for subtracting the mean on the numeric object from each observation.
The "scale" parameter (when set to TRUE) is responsible for dividing the resulting difference by the standard deviation of the numeric object.

REGRESSION ANALYSIS

Residual Analysis

SL No.	Temperature	% Yield	Predicted	Time
1	190	35.0	36.22	130
2	176	81.7	76.10	174
3	205	42.5	39.84	134
4	210	98.3	91.51	191
5	230	52.7	67.94	165
6	192	82.0	94.23	194
7	220	34.5	48.00	143
8	235	95.4	86.98	186
9	240	56.7	44.38	139
10	230	84.4	88.79	188
11	200	94.3	77.01	175
12	218	44.3	59.79	156
13	220	83.3	90.61	190
14	210	91.4	79.73	178
15	208	43.5	38.03	132
16	225	51.7	52.53	148

REGRESSION ANALYSIS

Step 5: Normality test using Shapiro Wilk Normality Test

```
> shapiro.test(Res)
```

Shapiro-Wilk normality Test:	
W	p value
0.9693	0.3132

Outlier test

Observations with Bonferonni p – value < 0.05 are potential outliers

```
> library(car)
> outlierTest(model_m)
```

Observation	Studentized Residual	Bonferonni p value
11	1.781515	NA

REGRESSION ANALYSIS

Leave One Out Cross Validation (LOOCV)

- Split the data into two parts : training data and test data
- Test data consists of only one observation (x_1, y_1)
- Training data consists of the remaining n – 1 observations namely (x_2, y_2),(x_3, y_3) , - - -, (x_n, y_n)
- Develop the model using n – 1 training data observations and predict the response y_1 of the test data observation
- Compute the residuals and mean square error $MSE_1 = (y_{1actual} - y_{1pred})^2$
- Repeat the process by taking (x_1, y_1) as test data and the remaining n – 1 observations as training data
- Compute MSE_2
- Repeating the procedure n times produces n squared errors MSE_1, MSE_2, - - -, MSE_n
- LOOCV estimate of the test MSE is the average of these n test error estimates

$$CV_{(n)} = \frac{1}{n}\sum_{i=1}^{n} MSE_i$$

REGRESSION ANALYSIS

Leave One Out Cross Validation (LOOCV)

```
> library(boot)
> attach(mydata)
> mymodel = glm(X.Yield ~ Time)
> myvalidation = cv.glm(mydata, mymodel)
> myvalidation$delta[1]
```

Statistic	Value
Delta	128.8541

STEPWISE REGRESSION

- Many predictor variables or independent variables ‘X_1, X_2, ...X_k’ (e.g.: gender, height) and a response variable or dependent variable ‘Y’ (e.g.: weight).

- It begins by selecting the single independent variable (entire set of predictors) that is the ‘best’ predictor which maximizes R^2. Then it adds (eliminates) variables in sequential manner, in order of importance and at each step it increases R^2.

- When you choose the stepwise method, you can enter a starting set of predictor variables in Predictors in initial model. These variables are removed if their p-values are greater than the Alpha to enter value. If you want keep variables in the model regardless of their p-values, enter them in Predictors to include in every model in the main dialog box.

BEST SUBSETS REGRESSION

- Many predictor variables or independent variables ‘$X_1, X_2, \ldots X_k$’ (e.g.: gender, height) and a response variable or dependent variable ‘Y’ (e.g.: weight).

- It generates regression models using the maximum R^2 criterion by first examining all one-predictor regression models and then selecting the two-predictor models giving the largest R^2. It examines all two-predictor models, selects the two models with the largest R^2, and displays information on these two models. This process continues until the model contains all predictors.
- Cp = (SSEp / MSEm) - (n-2p) : where SSEp is SSE for the best model with ‘p’ parameter and MSEm is the mean square error for the model with all ‘m’ predictors.
- We look for models where Cp is small and is also close to p, the number of parameters in the model.

Shrinkage Methods

Shrinkage Methods

The necessity to shrink

- Multicollinearity: As discussed earlier, as p (number of features) increases we are more likely to capture multiple features that have some multicollinearity. When multicollinearity exists, we often see high variability in our coefficient terms.
- Insufficient solution: When the number of features exceed the number of observations (p>n), the matrix $X^T X$ is not invertible. This causes significant issues, as:
    - The least-squares estimates are not unique. In fact, there are an infinite set of solutions available and most of these solutions overfit the data.
    - In many instances the result will be computationally infeasible.
- Interpretability: With a large number of features, we often would like to identify a smaller subset of these features that exhibit the strongest effects.
- In essence, we sometimes prefer techniques that provide feature selection. One approach to this is called hard threshholding feature selection, which can be performed with linear model selection approaches.
- However, model selection approaches can be computationally inefficient, do not scale well, and they simply assume a feature as in or out.
- We may wish to use a soft threshholding approach that slowly pushes a feature's effect towards zero.

Shrinkage Methods

Ridge regression and *Lasso*

- The subset selection methods use least squares to fit a linear model that contains a subset of the predictors.
- As an alternative, we can fit a model containing all p predictors using a technique that *constrains* or *regularizes* the coefficient estimates, or equivalently, that *shrinks* the coefficient estimates towards zero.
- It may not be immediately obvious why such a constraint should improve the fit, but it turns out that shrinking the coefficient estimates can significantly reduce their variance.

Ridge regression

- Recall that the least squares fitting procedure estimates $\beta_{0,}\beta_{1,} \dots, \beta_p$ using the values that minimize

$$RSS = \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 .$$

- In contrast, the ridge regression coefficient estimates $\widehat{\beta}^R$ are the values that minimize

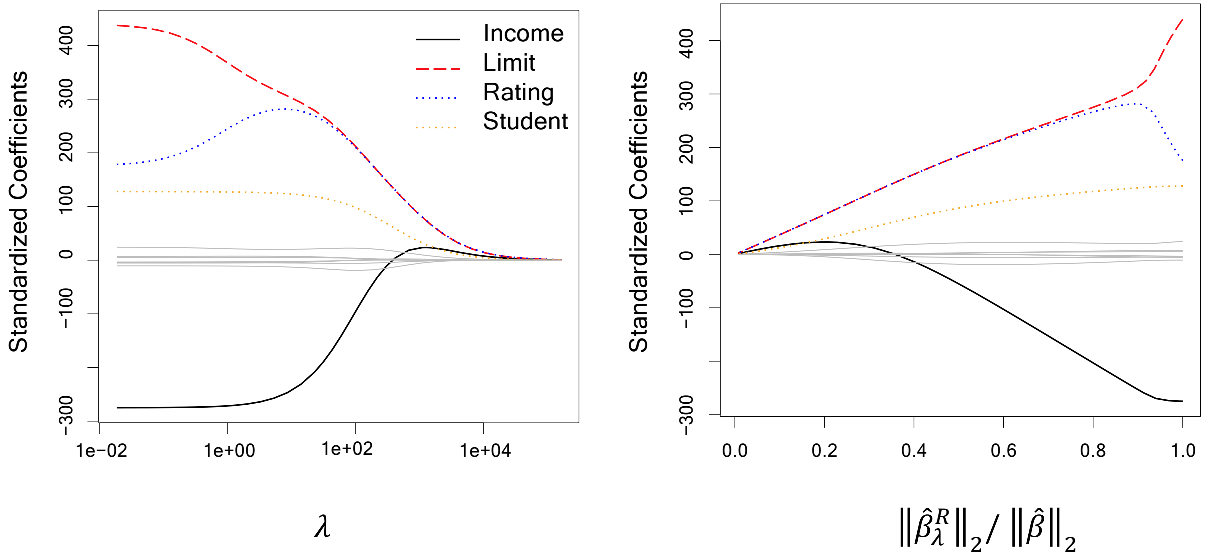
$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2 = RSS + \lambda\sum_{j=1}^{p}\beta_j^2,$$

where $\lambda \geq 0$ is a *tuning parameter*, to be determined separately.

Ridge regression

- As with least squares, ridge regression seeks coefficient estimates that fit the data well, by making the RSS small.
- However, the second term, $\lambda \sum_{j=1}^{p} \beta_j^2$, called a *shrinkage penalty*, is small when $\beta_{1,} \dots, \beta_p$ are close to zero, and so it has the effect of *shrinking* the estimates of β_j towards zero.
- The tuning parameter λ serves to control the relative impact of these two terms on the regression coefficient estimates.
- Selecting a good value for λ is critical; cross-validation is used for this.

Credit data example



Details of Previous Figure

- In the left-hand panel, each curve corresponds to the ridge regression coefficient estimate for one of the ten variables, plotted as a function of λ.
- The right-hand panel displays the same ridge coefficient estimates as the left-hand panel, but instead of displaying λ on the x-axis, we now display $\|\hat{\beta}_{\lambda}^{R}\|_2 / \|\hat{\beta}\|_2$,where $\hat{\beta}$ denotes the vector of least squares coefficient estimates.
- The notation $\|\beta\|_2$ denotes the l_2 norm (pronounced "ell 2") of a vector, and is defined as $\|\beta\|_2 = \sqrt{\sum_{j=1}^{p} \beta_j^2}$.

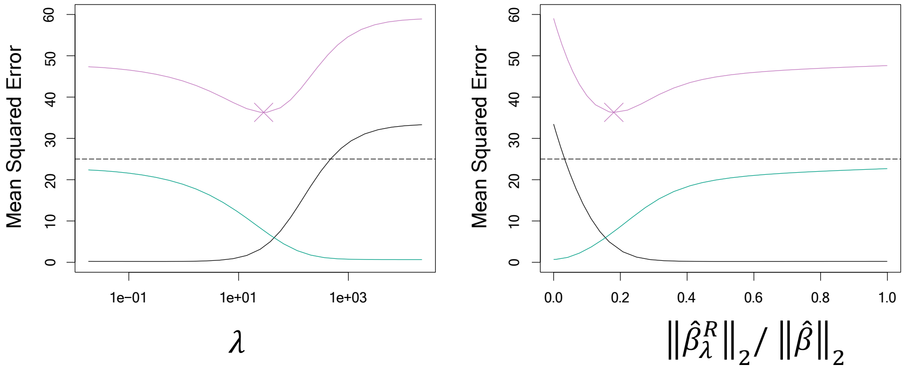
Ridge regression: Scaling of Predictors

- The standard least squares coefficient estimates are *scale equivariant*: multiplying X_j by a constant c simply leads to a scaling of the least squares coefficient estimates by a factor of $1/c$. In other words, regardless of how the jth predictor is scaled, $X_j\widehat{\beta}_j$ will remain the same.
- In contrast, the ridge regression coefficient estimates can change *substantially* when multiplying a given predictor by a constant, due to the sum of squared coefficients term in the penalty part of the ridge regression objective function.
- Therefore, it is best to apply ridge regression after *standardizing the predictors*, using the formula

$$\tilde{x}_{ij} = \frac{x_{ij}}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(x_{ij} - \bar{x}_j\right)^2}}$$

Why Does Ridge Regression Improve Over Least Squares?

The Bias-Variance tradeoff



Simulated data with $n =$ 50 observations, $p =$ 45 predictors, all having nonzero coefficients. Squared bias (black), variance (green), and test mean squared error (purple) for the ridge regression predictions on a simulated data set, as a function of λ and $\|\hat{\beta}_\lambda^R\|_2/\|\hat{\beta}\|_2$. The *horizontal dashed lines indicate the minimum possible MSE. The purple crosses indicate the ridge regression models for which the MSE is smallest.*

The Lasso

- Ridge regression does have one obvious disadvantage: unlike subset selection, which will generally select models that involve just a subset of the variables, ridge regression will include all *p* predictors in the final model.
- The *Lasso* is a relatively recent alternative to ridge regression that overcomes this disadvantage. The lasso coefficients, $\hat{\beta}_\lambda^L$,minimize the quantity

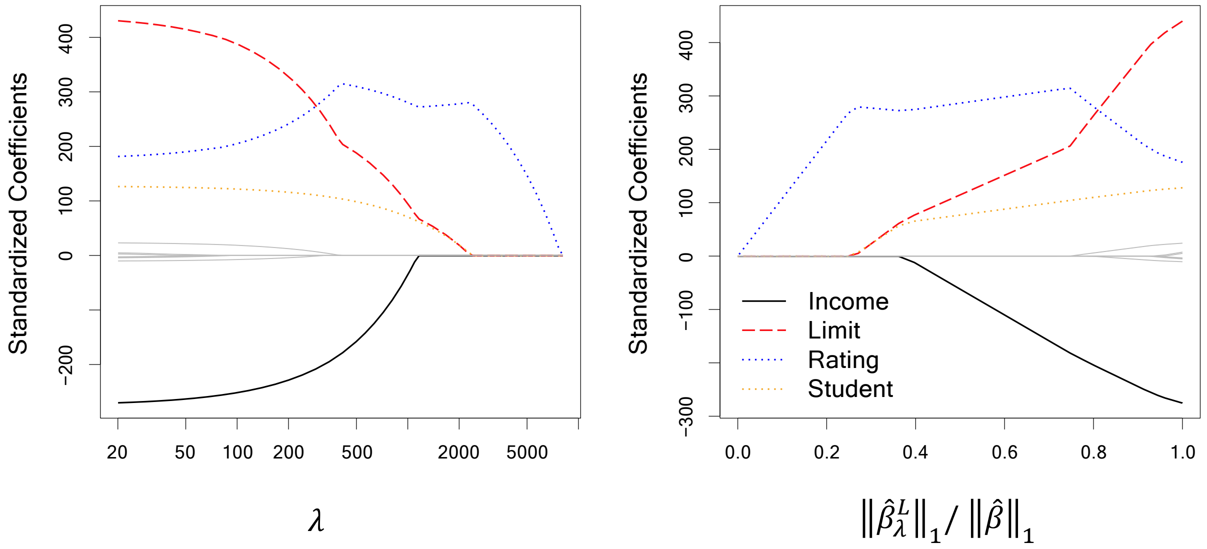
$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j| = RSS + \lambda\sum_{j=1}^{p}|\beta_j|,$$

- In statistical parlance, the lasso uses an l_1 (pronounced "ell 1") penalty instead of an l_2 penalty. The l_1 norm of a coefficient vector β is given by $\|\beta\|_1 = \sum_{j=1}^{p}|\beta_j|$.

The Lasso: continued

- As with ridge regression, the lasso shrinks the coefficient estimates towards zero.
- However, in the case of the lasso, the l_1 penalty has the effect of forcing some of the coefficient estimates to be exactly equal to zero when the tuning parameter λ is sufficiently large.
- Hence, much like best subset selection, the lasso performs *variable selection*.
- We say that the lasso yields *sparse* models — that is, models that involve only a subset of the variables.
- As in ridge regression, selecting a good value of λ for the lasso is critical; cross-validation is again the method of choice.

Example: Credit dataset



The Variable Selection Property of the Lasso

Why is it that the lasso, unlike ridge regression, results in coefficient estimates that are exactly equal to zero?

One can show that the lasso and ridge regression coefficient estimates solve the problems.

$$minimize_{\beta} \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij}\right)^2 \ subject\ to \sum_{j=1}^{p}|\beta_j| \leq s$$

and

$$minimize_{\beta} \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij}\right)^2 \ subject\ to \sum_{j=1}^{p}{\beta_j}^2 \leq s$$

respectively.

The Lasso Picture

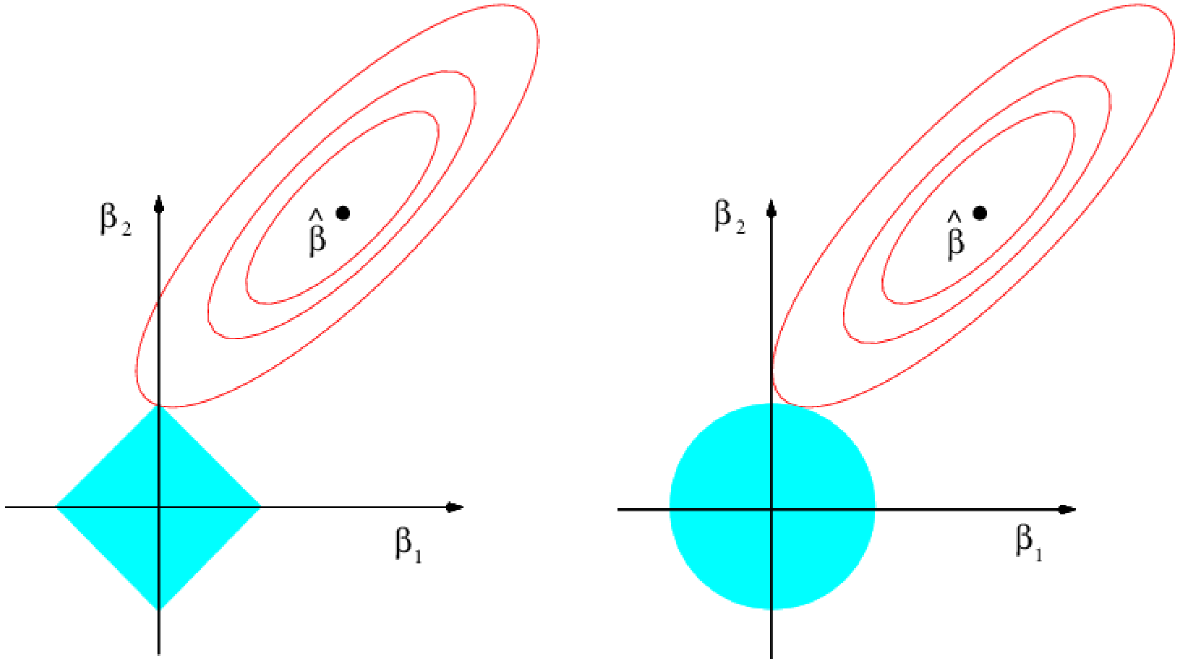
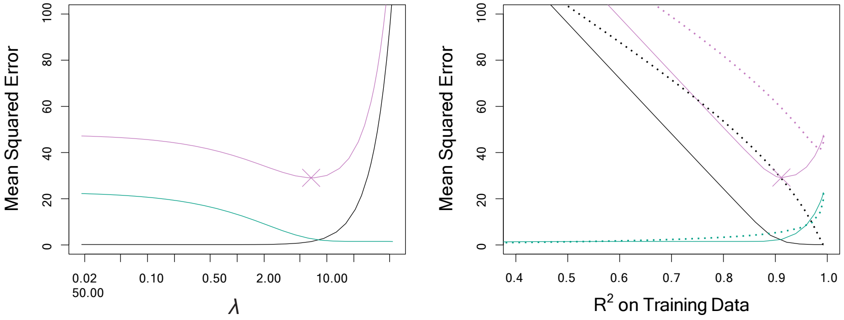


Figure 3.12: *Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions $|\beta_1| + |\beta_2| \leq t$ and $\beta_1^2 + \beta_2^2 \leq t^2$, respectively, while the red ellipses are the contours of the least squares error function.*

Comparing the Lasso and Ridge Regression: continued



Left: *Plots of squared bias (black), variance (green), and test MSE (purple) for the lasso. The simulated data is similar to that in previous slide* (Comparing the Lasso and Ridge Regression) *except that now only two predictors are related to the response.*
Right: *Comparison of squared bias, variance and test MSE between lasso (solid) and ridge (dashed). Both are plotted against their R^2 on the training data, as a common form of indexing. The crosses in both plots indicate the lasso model for which the MSE is smallest.*

Remarks

- These two examples illustrate that neither ridge regression nor the lasso will universally dominate the other.
- In general, one might expect the lasso to perform better when the response is a function of only a relatively small number of predictors.
- However, the number of predictors that is related to the response is never known *a priori* for real data sets.
- A technique such as cross-validation can be used in order to determine which approach is better on a particular data set.

Selecting the Tuning Parameter for Ridge Regression and Lasso

- As for subset selection, for ridge regression and lasso we require a method to determine which of the models under consideration is best.
- That is, we require a method selecting a value for the tuning parameter λ or equivalently, the value of the constraint s.
- *Cross-validation* provides a simple way to tackle this problem. We choose a grid of λ values, and compute the cross-validation error rate for each value of λ.
- We then select the tuning parameter value for which the cross-validation error is smallest.
- Finally, the model is re-fit using all of the available observations and the selected value of the tuning parameter.

Ridge Regression in R

We now illustrate the implementation of the two shrinkage methods in R.

We shall be working with the Ames Housing data, available in the *AmesHousing* package.

```
library(tidyverse)
library(modelr)
library(glmnet)
library(AmesHousing)
library(leaps)
library(broom)

# Create training and test data for Ames Housing data
dat <- make_ames()
set.seed(8885)
ames_split <- resample_partition(dat, c(test = 0.3, train = 0.7))
ames_train <- as_tibble(ames_split$train)
ames_test  <- as_tibble(ames_split$test)


# Create training and testing feature model matrices and response vectors.
# use model.matrix(...)[, -1] to discard the intercept
ames_train_x <- model.matrix(Sale_Price ~ ., ames_train)[, -1]
ames_train_y <- log(ames_train$Sale_Price)
ames_test_x <- model.matrix(Sale_Price ~ ., ames_test)[, -1]
ames_test_y <- log(ames_test$Sale_Price)
```

Ridge Regression in R

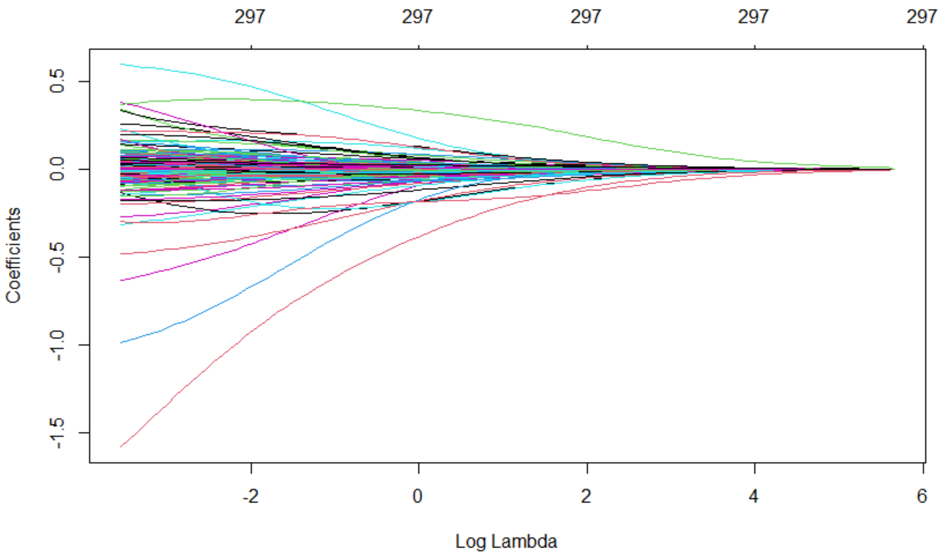
```
# Check dimensions
dim(ames_train_x)
## [1] 2052  308

dim(ames_test_x)
## [1] 878 308
```

We now apply the ridge regression method to the Ames Housing data.

```
# Apply Ridge regression to ames data
ames_ridge <- glmnet(x = ames_train_x,
                     y = ames_train_y, alpha = 0)

plot(ames_ridge, xvar = "lambda")
```



Ridge Regression in R

- In fact, we can see the exact λ values applied with *ames_ridge$lambda*.
- Although you can specify your own λ values, by default *glmnet* applies 100 λ values that are data derived. Majority of the time you will have little need to adjust the default λ values.
- We can also directly access the coefficients for a model using *coef* .
- *glmnet* stores all the coefficients for each model in order of largest to smallest λ.

```
# lambdas applied to penalty parameter
ames_ridge$lambda %>% head()
## [1] 288.3547 262.7381 239.3971 218.1298 198.7517 181.0952

# coefficients for the largest and smallest lambda parameters
coef(ames_ridge)[c("Gr_Liv_Area", "TotRms_AbvGrd"), 100]
##   Gr_Liv_Area TotRms_AbvGrd
##   0.000110669   0.010763563

coef(ames_ridge)[c("Gr_Liv_Area", "TotRms_AbvGrd"), 1]
##   Gr_Liv_Area TotRms_AbvGrd
##  5.727247e-40  1.334228e-37
```

Note how the larger λ values have pushed the coefficient estimates to nearly zero.

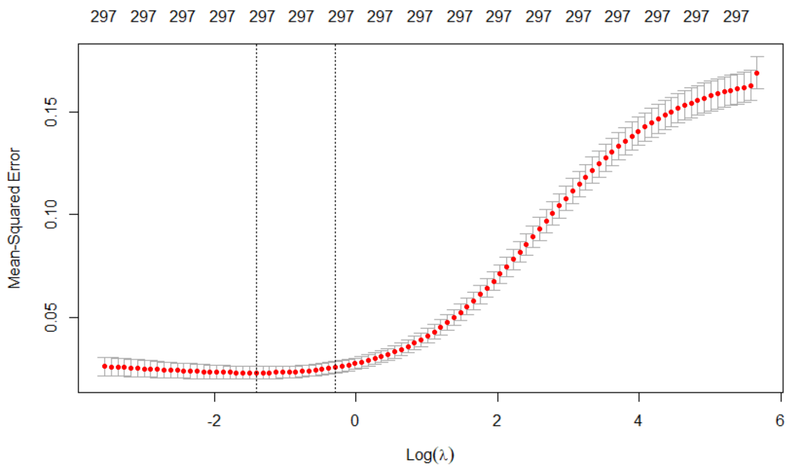
However, at this point, we do not understand how much improvement we are experiencing in our model.

Ridge Regression in R

Tuning the penalty parameter λ.

- Recall that λ is a tuning parameter that helps to control our model from over-fitting to the training data.
- However, to identify the optimal λ value we need to perform cross-validation (CV).
- *cv.glmnet* provides a built-in option to perform k-fold CV, and by default, performs 10-fold CV.

```
# Apply CV Ridge regression to ames data
ames_ridge <- cv.glmnet(x = ames_train_x, y = ames_train_y, alpha = 0)
# plot results
plot(ames_ridge)
```



Our plot output above illustrates the 10-fold CV mean squared error (MSE) across the λ values.

It illustrates that we do not see substantial improvement; however, as we constrain our coefficients with log(λ)>0, the MSE rises considerably.

Ridge Regression in R

The first and second vertical dashed lines represent the λ value with the minimum MSE and the largest λ value within one standard error of the minimum MSE.

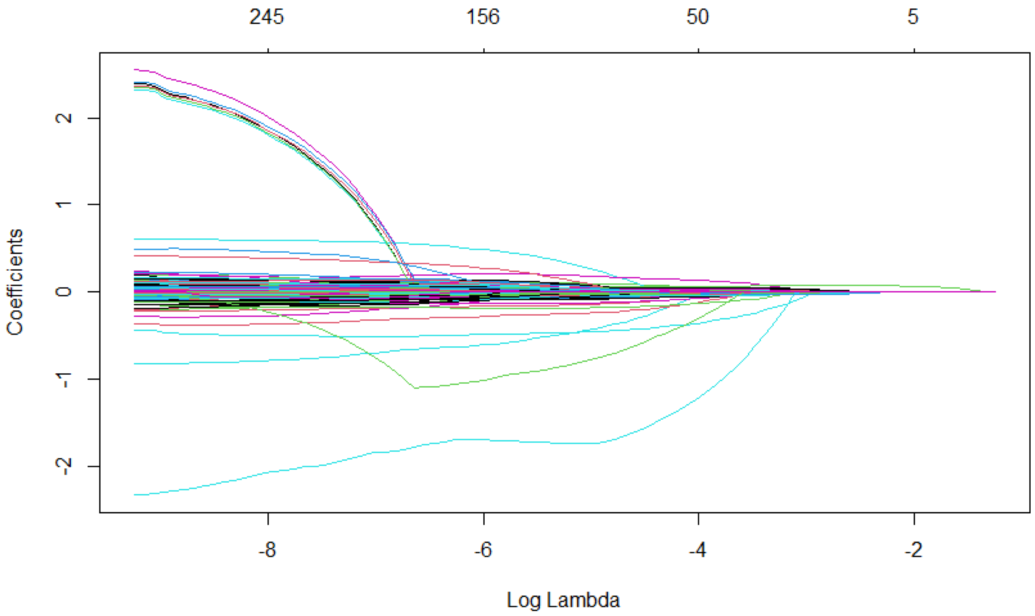
```
min(ames_ridge$cvm)       # minimum MSE
## [1] 0.02284662
ames_ridge$lambda.min     # lambda for this min MSE
## [1] 0.2450304
ames_ridge$cvm[ames_ridge$lambda == ames_ridge$lambda.1se]  # 1 st.error of min MSE
## [1] 0.02545676
ames_ridge$lambda.1se  # lambda for this MSE
## [1] 0.7482874
```

Prediction performance of the fitted ridge regression model.

```
ridge.pred <- predict(ames_ridge, s = ames_ridge$lambda.min, newx = ames_test_x)
mean((ridge.pred - ames_test_y)^2)
## [1] 0.01787904
```

LASSO Regression in R

We now apply the ridge regression method to the Ames Housing data.
ames_lasso <- glmnet(x = ames_train_x, y = ames_train_y, alpha = 1)
plot(ames_lasso, xvar = "lambda")



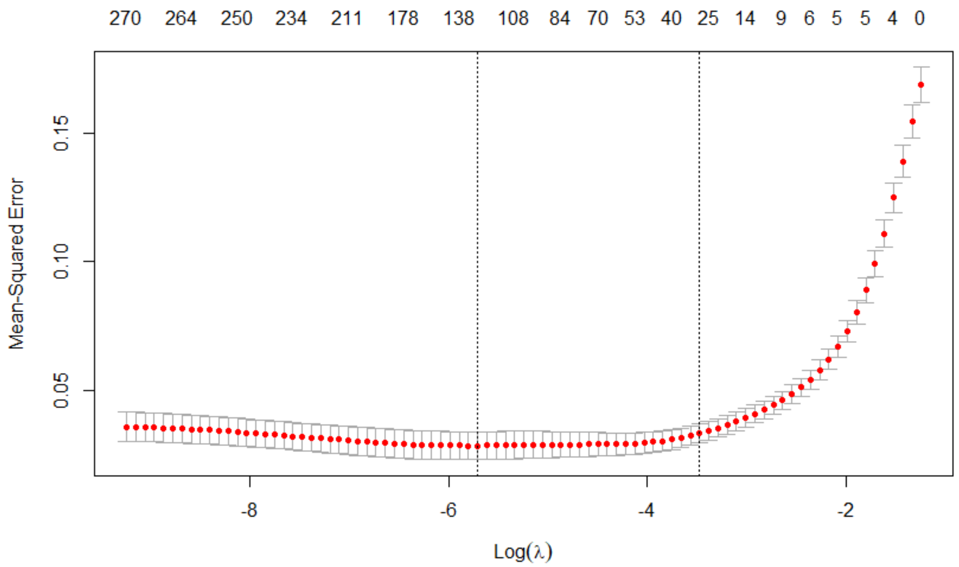
Tuning the penalty parameter λ.

# Apply CV lasso regression to ames data

ames_lasso <- cv.glmnet(x = ames_train_x, y = ames_train_y, alpha = 1)

LASSO Regression in R

```
# plot results
plot(ames_lasso)
```



```
min(ames_lasso$cvm)      # minimum MSE
## [1] 0.02828329

ames_lasso$lambda.min     # lambda for this min MSE
## [1] 0.003315378

ames_lasso$cvm[ames_lasso$lambda == ames_lasso$lambda.1se]  # 1st.error of min MSE
## [1] 0.03308106
```

LASSO Regression in R

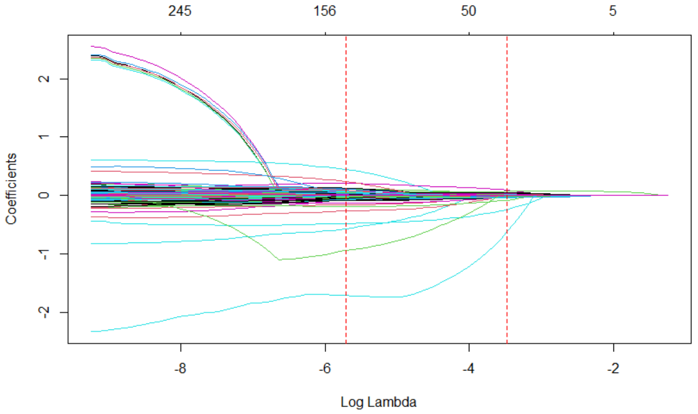
```
ames_lasso$lambda.1se  # lambda for this MSE
## [1] 0.03091933

#Prediction performance of the fitted lasso regression model.
lasso.pred <- predict(ames_lasso, s = ames_lasso$lambda.min, newx = ames_test_x)
mean((lasso.pred - ames_test_y)^2)
## [1] 0.02093244

lasso.pred2 <- predict(ames_lasso, s = ames_lasso$lambda.1se, newx = ames_test_x)
mean((lasso.pred2 - ames_test_y)^2)
## [1] 0.02726125

# Advantage of choosing λ with MSE within 1 standard error
ames_lasso_min <- glmnet(
  x = ames_train_x,
  y = ames_train_y,
  alpha = 1
)

plot(ames_lasso_min, xvar = "lambda")
abline(v = log(ames_lasso$lambda.min), col = "red", lty = "dashed")
abline(v = log(ames_lasso$lambda.1se), col = "red", lty = "dashed")
```



REGRESSION ANALYSIS – Another Example

Exercise : The effect of temperature, time and kappa number of pulp affects the % conversion of UB pulp to Cl_2 pulp. inspection. The data collected in given in the Mult_Reg_Conversion file. Develop a model for % conversion in terms of exploratory variables?

Step 1: Reading the data and variables

```
> data = read.csv(‘Mult_Reg_Conversion.csv',header = T,sep = ",")
> mydata[,-1]  # Removing 1st  column
> attach(mydata)
```

REGRESSION ANALYSIS

Step 1: Correlation Analysis

```
> cor(mydata)
```

	Temperature	Time	Kappa #	X..Conversion
Temperature	1.00	-0.96	0.22	0.95
Time	-0.96	1.00	-0.24	-0.91
Kappa #	0.22	-0.24	1.00	0.37
X..Conversion	0.95	-0.91	0.37	1.00

Interpretation

High Correlation between X..Conversion and Temperature & Time

High Correlation between Temperature & Time - Multicollinearity

REGRESSION ANALYSIS

Measure for Multicollinearity

Variance Inflation Factor (VIF)

Measures the correlation (linear association) between each x variable with other x's

$VIF_i = 1/(1- R_i^2)$

Where R_i is the coefficient for regressing x_i on other x's

Criteria: VIF > 5 indicates multicollinearity.

REGRESSION ANALYSIS

Regression Output

- model = lm(X..Conversion ~ Kappa.number + Temperature + Time)

```
> summary(model)
```

	Coeff	Std. Error	t	p value
Constant	-121.27	55.43571	-2.19	0.0492
Temperature	0.12685	0.04218	3.007	0.0109
Time	-19.0217	107.92824	-0.18	0.863
Kappa #	0.34816	0.17702	1.967	0.0728

Variance-inflation factors (VIF)

```
> vif(mymodel)
```

x	VIF
Temperature	12.23
Time	12.33
Kappa #	1.062

REGRESSION ANALYSIS

Regression Output

Statistic	Value	Criteria
Adjusted R Square	0.899	> 0.6

Regression ANOVA

> anova(model)

Model	SS	df	MS	F	p value
Kappa.number	290.79	1	290.79	20.4915	0.000694
Temperature	1662.19	1	1662.19	117.1310	0
Time	0.44	1	0.44	0.0311	0.8630417
Residual	170.290	12	14.191		
Total	2123.709	15			

REGRESSION ANALYSIS

Methods for Tackling Multicollinearity:

1. Remove one or more of highly correlated independent variable
2. Principal Component Regression
3. Partial Least Square Regression
4. Ridge Regression
5. Collecting Additional Data

REGRESSION ANALYSIS

Tackling Multicollinearity:

Method 1: Removing highly correlated variable – Stepwise Regression

Approach

- A null model is developed without any predictor variable x. In null model, the predicted value will be the overall mean of y
- Then predictor variables x's are added to the model sequentially
- After adding each new variable, the method also remove any variable that no longer provide an improvement in the model fit
- Finally the best model is identified as the one which minimizes Akaike information criterion (AIC)

$$AIC = \frac{1}{n\hat{\sigma}^2}(RSS + 2d\hat{\sigma}^2)$$

n: number of observations

$\hat{\sigma}^2$: estimate of error or residual variance

d: number of x variables included in the model

RSS: Residual sum of squares

REGRESSION ANALYSIS

Tackling Multicollinearity:

Method 1: Removing highly correlated variable – Stepwise Regression

R code

```
> library(MASS)
> mymodel = lm(X..Conversion ~ Temperature + Time + Kappa.number)
> step =stepAIC(mymodel, direction = "both")
```

Step	x's in the model	AIC
1	Temperature, Time & Kappa Number	45.8
2	Temperature & Kappa Number	43.9

REGRESSION ANALYSIS

Tackling Multi collinearity:

Method 1: Stepwise Regression

> summary(step)

Attribute	Coefficient	Std. Error	t Statistic	p value
Temperature	0.13396	0.01191	11.250	0.0000
Kappa #	0.35106	0.16955	2.071	0.0589
Intercept	-130.68986	14.14571	-9.239	0.0000

X..Conversion = 0.13396 * Temperature + 0.35106 * Kappa # - 130.68986

Variance-inflation factors (VIF)

> vif(step)

x	VIF
Temperature	1.0526
Kappa #	1.0526

REGRESSION ANALYSIS

Tackling Multi collinearity:

Method 1: Stepwise Regression

```
> pred = predict(step)
> res = residuals(step)
 > cbind(X..Conversion, pred, res)
> mse = mean(res^2)
> rmse = sqrt(mse)
```

Statistic	Value
Mean Square Error (MSE)	10.7
Root Mean Square Error (RMSE)	3.27

REGRESSION ANALYSIS

k fold Cross Validation

Steps

1. Divide the data set into k equal subsets
2. Keep one subset (sample) for model validation
3. Develop the model using all the other k – 1 subsets data put together
4. Predict the responses for the test data and compute residuals
5. Return the test sample back to the original data set and take another subset for model validation
6. Go to step 3 and continue until all the subsets are tested with different models
7. Compute the overall Root Mean Square Residuals. RMSE of validation should not be high compared to the original model developed with all the data points together.

Note: when k = n, then k fold cross validation is same as leave one out cross validation

REGRESSION ANALYSIS

k fold Cross Validation

R code

```
> library(DAAG)
> cv.lm(mymodel, m = 16)
> cv.lm(mymodel, df = mydata, m = 16)
```

m: number of validations required. M = 16 = n, hence equal to leave one out cross validation

Model	MSE	RMSE
Original	10.7	3.27
Cross Validation	19.6	4.43

REGRESSION ANALYSIS

Tackling Multi collinearity:

Method 2: Principal Component Regression

1.Perform principal component analysis on x variables
2.Use the principal components as x variables and develop the model

R Code : Principal Component Regression

```
>install.packages(“pls”)
> library(pls)
> mymodel = pcr(X..Conversion ~ ., data = mydata, scale = TRUE)
> summary(mymodel)
➢mymodel$loadings
➢mymodel$scores
```

REGRESSION ANALYSIS

Tackling Multi collinearity:

Method 2: Principal Component Regression

1.Perform principal component analysis on x variables
2.Use the principal components as x variables and develop the model

Cum % Variance	PC1	PC2	PC3
x	68.66	98.61	100
Conversion (y)	90.48	90.62	91.98

Component 1 or 1 & 2 may be sufficient to include in the model

REGRESSION ANALYSIS

Tackling Multi collinearity:

Method 2: Principal Component Regression

1.Perform principal component analysis on x variables
2.Use the principal components as x variables and develop the model

Loadings	PC1	PC2	PC3
Temperature	-0.674	0.218	0.705
Time	0.677	-0.2	0.709
Kappa.number	-0.296	-0.955	0

Component 1 is taking care of information in temperature and Time and Component 2 is mostly representing kappa number

REGRESSION ANALYSIS

Tackling Multi collinearity:

Method 2: Principal Component Regression

Principal Component Scores

SL No.	Comp 1	Comp 2	Comp 3
1	-1.079	1.2498	0.1202
2	-1.158	0.9967	0.1236
3	-1.273	0.6625	0.117
4	-1.371	0.2313	0.1563
5	-1.543	-0.362	0.1756
6	-1.889	-1.365	0.1558
7	0.4709	1.1733	-0.133
8	0.3133	0.8148	-0.173
9	0.0021	0.2622	-0.299
10	-0.257	-0.122	-0.428
11	-0.268	-0.763	-0.24
12	-0.432	-1.819	-0.07
13	2.2484	0.6246	-0.022
14	2.4329	0.165	0.2963
15	2.1218	-0.388	0.1699
16	1.6801	-1.362	0.0493

REGRESSION ANALYSIS

Tackling Multi collinearity:

Method 2: Principal Component Regression

Identifying the required number of components in the model

```
> pred = predict(mymodel, type = "response", ncomp = 1)
> res = X..Conversion - pred
> mse = mean(res^2)
> prednew = predict(mymodel, type = "response", ncomp = 2)
> resnew = X..Conversion - prednew
> msenew = mean(resnew^2)
```

Statistics	Regression with	
	PC1	PC1 & PC2
MSE	12.64226	12.45593

Since there is not much reduction in MSE by including the second principal component , only PC1 is required for modelling

REGRESSION ANALYSIS

Tackling Multi collinearity:

Method 3: Partial Least Square Regression

Principal component regression involves the identification of a linear combinations of predictors that best represents the x variables

The response y is not used to help the determination of principal components

The response does not supervise the identification of principal components

Identifies the best linear combinations which best explains the predictor variables x but may not the ones best for predicting the response y

Partial least square regression is a supervised alternative to principal component regression

Partial least square method identifies the components or directions (linear combinations of x variables) using the response variable y.

Partial least square places highest weight on the variables that are most strongly related the response y

REGRESSION ANALYSIS

Tackling Multi collinearity:

Method 3: Partial Least Square Regression

R code

```
> mymodel = plsr(X..Conversion ~ ., data = mydata, scale = TRUE)
> summary(mymodel)
> mymodel$loading
```

REGRESSION ANALYSIS

Tackling Multi collinearity:

Method 3: Partial Least Square Regression

Cum % Variance	PLS1	PLS2	PLS3
x	68.65	96.92	100
Conversion (y)	90.63	90.86	91.98

Loadings	PLS1	PLS2	PLS3
Temperature	0.677	0.344	0.299
Time	-0.679	-0.207	0.607
Kappa.number	0.285	-1.391	0.736

REGRESSION ANALYSIS

Tackling Multi collinearity:

Method 3: Partial Least Square Regression

```
> ps = mymodel$scores
> score = ps[,1:2]
```

SL No	PLS1	PLS2
1	1.11324	0.89634
2	1.18502	0.73368
3	1.2913	0.51027
4	1.3792	0.25877
5	1.5361	-0.1142
6	1.85493	-0.7845
7	-0.4425	0.66627
8	-0.2949	0.40157
9	-0.0005	-0.0564
10	0.24599	-0.4059
11	0.24426	-0.6809
12	0.3833	-1.24
13	-2.2314	0.4067
14	-2.4222	0.35105
15	-2.1279	-0.1069
16	-1.7138	-0.8359

REGRESSION ANALYSIS

Tackling Multi collinearity:

Method 3: Partial least square regression

Identifying the required number of components in the model

```
> pred = predict(mymodel, data = mydata, scale = TRUE, ncomp = 1)
> res = X..Conversion - pred
> mse = mean(res^2)

> prednew = predict(mymodel, , data = mydata, scale = TRUE ,  ncomp = 2)
> resnew = X..Conversion - prednew
> msenew = mean(resnew^2)
```

Statistics	Regression with	
	PLS1	PLS11 & PLS2
MSE	12.44252	12.13185

Since there is not much reduction in MSE by including the second component , only PLS1 is required for modelling

REGRESSION ANALYSIS

Tackling Multicollinearity:

Method 4: Ridge regression

In least square regression, the coefficients β's of x variables are identified by minimizing the residual sum of squares

$$RSS = \sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2$$

In ridge regression, the coefficients β's of x variables are identified by minimizing a slightly different quantity

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2 + \lambda\sum_{j=1}^{p}\beta_j^2 = RSS + \lambda\sum_{j=1}^{p}\beta_j^2$$

Where λ ≥ 0 is a turning parameter and $\lambda\sum_{j=1}^{p}\beta_j^2$ is the shrinkage penalty,

which will be small when β_1, β_2, - - -, β_p are close to zero.

REGRESSION ANALYSIS

Tackling Multi collinearity:

Method 4: Ridge regression

Ridge regression seeks coefficient estimates that fit the data well by minimizing the RSS and the tuning parameter λ has the effect of shrinking the estimates β_j towards zero
The value of λ s identified through 10 fold cross validation

10 fold Cross Validation

- Divide the data set into 10 equal parts
- Develop the model using 9 parts and test it with the remaining one part
- Repeat the process 10 times to get an unbiased estimate of MSE

REGRESSION ANALYSIS

Tackling Multi collinearity:

Method 4: Ridge regression

R Code

```
> library(glmnet)
> set.seed(1)
> y = mydata[,4]
> x =mydata[,1:3]
> x = as.matrix(x)
```

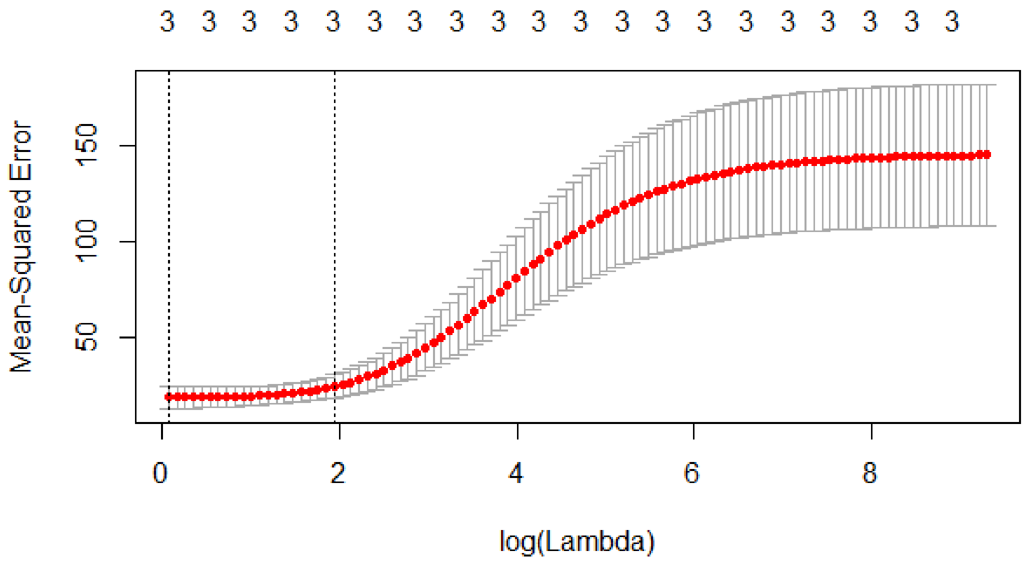
Cross Validation

```
> mymodel = cv.glmnet(x , y, alpha =0)
> plot(mymodel)
```

REGRESSION ANALYSIS

Tackling Multi collinearity:

Method 4: Ridge regression



Choose the λ which minimizes the mean square error

> bestlambda = mymodel$lambda.min

Best λ= 1.088771

REGRESSION ANALYSIS

Tackling Multi collinearity:

Method 4: Ridge regression

Develop the model with best λ and identify the coefficients

```
> mynewmodel = glmnet(x, y, alpha = 0)
> predict (mynewmodel, type = "coefficients", s = bestlambda)[1:4,]
```

Variable	Coefficients
(Intercept)	-63.0713
Temperature	0.0823
Time	-117.5048
Kappa.number	0.3268

Regression with dummy variables

When x’s are not numeric but nominal

Each nominal or categorical variable is converted into dummy variables

Dummy variables takes values 0 or 1

Number of dummy variable for one x variable is equal to number of distinct values of that variable - 1

Example: A study was conducted to measure the effect of gender and income on attitude towards vocation. Data was collected from 30 respondents and is given in Travel_dummy_reg file. Attitude towards vocation is measured on a 9 point scale. Gender is coded as male = 1 and female = 2. Income is coded as low=1, medium = 2 and high = 3. Develop a model for attitude towards vocation in terms of gender and Income?

CORRELATION & REGRESSION

Regression with dummy variables

Variable		Dummy
Gender	Code	gender_Code
Male	1	0
Female	2	1

Variable		Dummy	
Income	Code	Income1	Income 2
Low	1	0	0
Medium	2	1	0
High	3	0	1

CORRELATION & REGRESSION

Regression with dummy variables

Read the fie and variables

➢mydata = Travel_dummy_Reg

➢attach(mydata)

> mydata = mydata[,2:4]

Converting categorical x's to factors

> gender = factor(Gender)

> income = factor(Income)

CORRELATION & REGRESSION

Regression with dummy variables – Output

```
> mymodel =  lm(Attitude ~ gender + income)
> summary (mymodel)
```

Multiple R^2	0.8603
Adjusted R^2	0.8442
F Statistics	53.37
P value	0.00

	Estimate	Std. Error	t value	p value
(Intercept)	2.4	0.3359	7.145	0.00000
gender2	-1.6	0.3359	-4.763	0.00006
income2	2.8	0.4114	6.806	0.00000
income3	4.8	0.4114	11.668	0.00000

CORRELATION & REGRESSION

Regression with dummy variables – Output

```
> anova (mymodel)
```

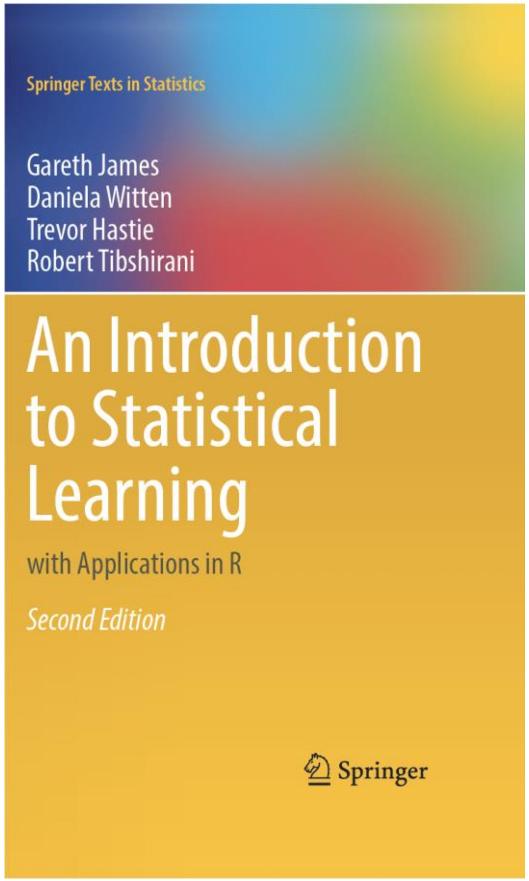
	Df	Sum Sq	Mean Sq	F	p value
gender	1	19.2	19.2	22.691	0.0001
income	2	116.27	58.133	68.703	0.0000
Residuals	26	22	0.846		

Exercise Problem

Apply Ridge Regression and Lasso approach to the Hitters data from ISLR2 package in R and to predict a basketball player's Salary on the basis of various features

CHEAT SHEET

X variable	Y variable	Regression type
Numeric	Numeric	OLS
Categorical	Numeric	OLS with dummy
Numeric	Categorical	Logistic regression
Categorical	Categorical	Logistic regression with dummy



Springer Texts in Statistics

Gareth James
Daniela Witten
Trevor Hastie
Robert Tibshirani

An Introduction to Statistical Learning

with Applications in R

Second Edition

Springer