# CWRU DSCI351-451: Intro to LinRegr-ISLR2

*Roger H. French, JiQi Liu*

*05 November, 2018*

## Contents





#### 10.2.1.1 Reading, Homeworks, Projects, SemProjects

- Homework:
  - HW6 is given out Thursday November 8th, 2018
- Readings:
  - ISLR Chapter 3
- Projects: We will have four 2 week EDA projects
  - Project 3 given out today, November 6th, 2018
  - Due Thursday November 20, 2019
- 451 SemProjects:
  - Third SemProj Report Outs Dec. 4,6 2018
  - Final full SemProject Written Report Due 12/17/2018
- Final Exam
  - Monday December 17th, 12 noon to 3pm, Olin 313

#### 10.2.1.2 Syllabus

| Day:Date | Foundation | Practicum | Reading | Due |
|---|---|---|---|---|
| w1a:Tu:8/28/18 | ODS Tool Chain | R, Rstudio, Git | | |
| w1b:Th:8/30/18 | Setup ODS Tool Chain | Bash, Git, Twitter | PRP4-33 | HW1 |
| w2a:Tu:9/4/18 | What is Data Science | OIS:Intro2R | PRP35-64 | **HW1 Due** |
| w2b:Th:9/6/18 | Data Analytic Style, Git | 451SempProj, Git | PRP65-93, OI1-1.9 | HW2 |
| w3a:Tu:9/11/18* | Struct. of Data Analysis | ISLR:Intro2R, Loops | PRP94-116, OIS3 | **HW2 Due** |
| w3b:Th:9/13/18* | OIS3 Intro to Data | GapMinder, Dplyr, Magrittr | | |
| w4a:Tu:9/18/18 | OIS3, Intro2Data part 2, Data | EDA: PET Degr. | EDA1-31 | Proj1 |
| w4b:Th:9/20/18 | Hypothesis Testing | GGPlot2 Tutorial | EDA32-58 | HW3 |
| w5a:Tu:9/25/18 | Distributions | SemProj RepOut1 | R4DS1-3 | **HW3 Due** |
| w5b:Th:9/27/18 | Wickham DSCI in Tidyverse | SemProj RepOut1 | R4DS4-6 | **SemProj1,** |
| w6a:Tu:10/2/18 | OIS Found. of Inference | Inference | R4DS7-8 | **Proj1 Due** |
| w6b:Th:10/4/18 | | Midterm Review | R4DS9-16 Wrangle | |
| w7a:Tu:10/9/18* | Summ. Stats & Vis. | Data Wrangling | | |
| w7b:Th:10/11/18* | **MIDTERM EXAM** | | | HW4 |
| w8a:Tu:10/16/18 | Numerical Inference | Tidy Check Explore | OIS4 | **HW4 Due** |
| w8b:Th:10/18/18 | Algorithms, Models | Pairwise Corr. Plots | OIS5.1-4 | Proj 2, HW5 |
| Tu:10/23 | **CWRU FALL BREAK** | | R4DS17-21 Program | |
| w9b:Th:10/25/18 | Categorical Infer | Predictive Analytics | OIS6.1,2 | |
| w10a:Tu:10/30/18 | SemProj | SemProj | OIS7 | **SemProj2 HW5 Du** |
| w10b:Th:11/1/18 | Lin. Regr. | Lin. Regr. | OIS8 | **Proj.2 due** |
| w11a:Tu:11/6/18 | Inf. for Regression | Curse of Dim. | OIS8 | Proj 3 |
| w11b:Th:11/8/18 | Model Accuracy | Training Testing | ISLR3 | HW6 |
| w12a:Tu:11/13/18 | Multiple Regr. | Mul. Regr. & Pred. | ISLR4 | **HW6 due** |
| w12b:Th:11/15/18 | Classification | | ISLR6 | |
| w13a:Tu:11/20/18 | Classification | Clustering | ISLR5 | **Proj 3 due** |
| Th:11/22/18 | **THANKSGIVING** | | | Proj 4 |
| w14a:Tu:11/27/18 | Big Data | Hadoop | | |
| w14b:Th:11/29/18 | InfoSec | VerisDB | | **SemProj3** |
| w15a:Tu:12/4/18 | SemProj ReportOut3 | | | |
| w15b:Th:12/6/18 | SemProj ReportOut3 | | | **Proj4** |
| | **FINAL EXAM** | **Monday12/17, 12:00-3:00pm** | Olin 313 | **SemProj4 due** |

Figure 1: DSCI351/451 Syllabus

#### 10.2.1.3 ISLR Chapter 2 Regression and IntroR Lab Excerise

From Hastie and Tibshirani

- They have good notation
- And a good intro to R

##### 10.2.1.3.1 Regression is the case of supervised learning

Where we have a response that is associated with the predictors

- And we want to develop a predictive model
  - that relates predictors with response

#### 10.2.1.4 Function Notation for a Predictive model

Some notation for predictive models

- Response $Y$ which we want to predict
- And the Predictors we will use are $X = X_1 + X_2 + X_3$
  - when we have $P$ number of predictors,
  - and $P = 3$ in this example
  - where the predictors $X$ is a vector
  - And $X$ is a column vector containing $(X_1, X_2, X_3)$
  - Which has 3 components $X_1 + X_2 + X_3$
  - We also have to have an error term $\epsilon$
- Our predictive model will then be
  - $Y = f(X) + \epsilon$
- $\epsilon$ error term is a catch all
  - captures measurement error, and other discrepancies
  - we can never model something perfectly
- And for the predictor $X$
  - A single instance of $X$ is $x$
  - i.e. $(x_1, x_2, x_3)$
  - three specific values of the 3 components
  - of 1 individual observation, i.e. $x$
  - of the predictor $X$

##### 10.2.1.4.1 Variables

- Independent Variables $X$ are called
  - independent variables
  - predictors
  - exogenous variables
- Dependent Varialbes $Y$ are called
  - dependent variables
  - responses
  - endogenous variables

In some cases, such as network models

- Some variables may be both.
  - independent, predictors
  - and also dependent response
- Such as in our group's netSEM structural equation models
  - take a look at SEM package

```r
# install.packages("sem")
library(sem)
help(sem)


# install.packages("lavaan")
library(lavaan)
```

```
## This is lavaan 0.6-3

## lavaan is BETA software! Please report any bugs.

##
## Attaching package: 'lavaan'

## The following objects are masked from 'package:sem':
##
##     cfa, sem
```

```r
help(lavaan)

# install.packages("netSEM")
library(netSEM)
help(netSEM)
```

##### 10.2.1.4.2 Expected Values of a Predictive Model

Now, once you have a predictive model

How well does it do, fitting your actual response?

- Remember a function is by definition single-valued
  - for a given value $x_1$ of the independent variable X
  - there is only dependent value $y_1$ for the dependent variable Y
- Therefore it can never actually predict
  - the exact observed value of the response
- this is why we keep the error term $\epsilon$ explicit

The Expected Value of a Regression Function

- Our regression function is $Y = f(X) + \epsilon$
- Gives the Expected value of the response for $X = 4$

Notation for this is:

$$f(4) = E(Y|X = 4)$$

Or for our vector $X$

$$f(x) = f(x_1, x_2, x_3) = E(Y|X_1 = x_1, X_2 = x_2, X_3 = x_3)$$

##### 10.2.1.4.3 The ideal or optimal predictor of Y

- Minimizes the loss function between the function and the data
- For example minimizing the sum of squared errors

## The regression function $f(x)$

- Is also defined for vector $X$; e.g. $f(x) = f(x_1, x_2, x_3) = E(Y|X_1 = x_1, X_2 = x_2, X_3 = x_3)$
- Is the *ideal* or *optimal* predictor of $Y$ with regard to mean-squared prediction error: $f(x) = E(Y|X = x)$ is the function that minimizes $E[(Y - g(X))^2|X = x]$ over all functions $g$ at all points $X = x$.
- $\epsilon = Y - f(x)$ is the *irreducible* error — i.e. even if we knew $f(x)$, we would still make errors in prediction, since at each $X = x$ there is typically a distribution of possible $Y$ values.
- For any estimate $\hat{f}(x)$ of $f(x)$, we have

$$E[(Y - \hat{f}(X))^2|X = x] = \underbrace{[f(x) - \hat{f}(x)]^2}_{\textit{Reducible}} + \underbrace{\mathrm{Var}(\epsilon)}_{\textit{Irreducible}}$$

Figure 2: the regression function and its nature

#### 10.2.1.4.4 An estimate (one version) of $f(X)$

- is called $\hat{f}(X)$
- since we could determine many versions of $f(X)$

And then we'll determine the best one of these $\hat{f}(X)$ functions

- That reduces the loss function

#### 10.2.1.4.5 And then we are left with the irreducible error

- Which is just the variance of the errors.

#### 10.2.1.4.6 So by better model building

- we can reduce the reducible error
- and we're left with the irreducible error.
  - Which I think of as the true "noise" in the data

### 10.2.1.5 Overview of the Regression Function and its nature

##### 10.2.1.5.1 How do we estimate the function $f(X)$?

We can perform the loss function minimization, at each specific value $x$ of $X$.

- Or at least in the neighborhood of $x$,
  - which is denonte by $\mathcal{N}(x)$
  - and called Nearest Neighbor Averaging

Note that the regression function $f(X)$ is not an algebraic function

- We didn't guessestimate it should be quadratic or some such.
- It is a numerical function defined for each value $x$ of $X$

#### 10.2.1.6 The Curse of Dimensionality

When we are doing our nearest neighborhood averaging

- in high dimensional datasets
- we are hit by the curse of dimensionality
  - We can't define who are nearest neighbors
  - Because they tend to be far away in high dimensions

This hits us in many places of Prediction, Modeling and Statistical Learning

- The Curse of Dimensionality

#### 10.2.1.7 Parametric and Structured Models

One way to get around the curse of dimensionality,

- Use Parametric Models

$$f_l(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ...\beta_p X_p$$

Where there are $p + 1$ parameters in the model

- Which are estimated by fitting the model to the data

Estimated values of a parameter $\beta$

- are denoted as $\hat{\beta}$

##### 10.2.1.7.1 Some tradeoffs in regression modeling

- Prediction accuracy versus interpretability.
  - Linear models are easy to interpret;
  - thin-plate splines are not.
- Good fit versus over-fit or under-fit.
  - How do we know when the fit is just right?
- Parsimony versus black-box.
  - We often prefer a simpler model
  - involving fewer variables
  - Over a black-box predictor
  - involving them all.

# How to estimate $f$

- Typically we have few if any data points with $X = 4$ exactly.
- So we cannot compute $E(Y|X = x)$!
- Relax the definition and let

$$\hat{f}(x) = \text{Ave}(Y|X \in \mathcal{N}(x))$$

where $\mathcal{N}(x)$ is some *neighborhood* of $x$.

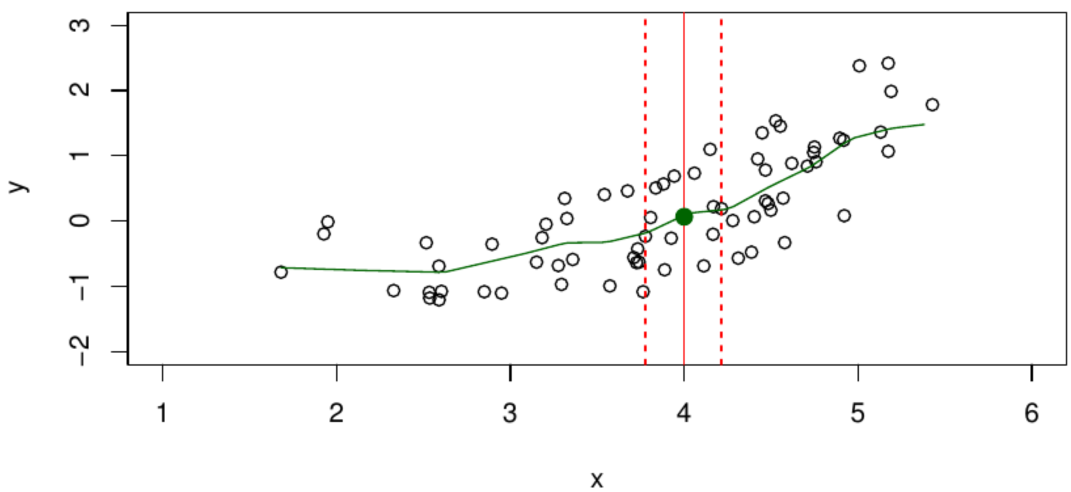
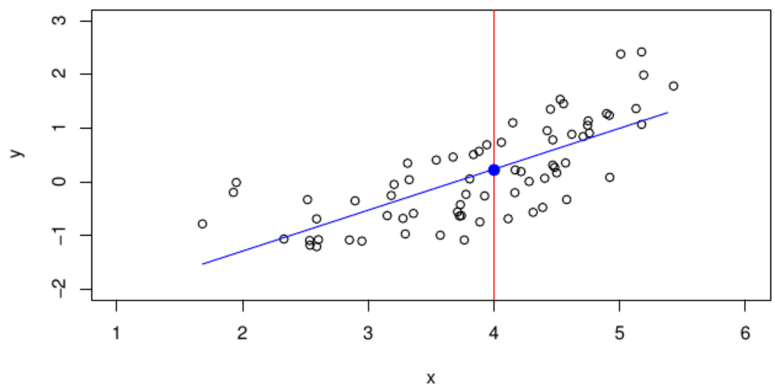



Figure 3: how to determine the regression function $f(X)$

A linear model $\hat{f}_L(X) = \hat{\beta}_0 + \hat{\beta}_1 X$ gives a reasonable fit here




A quadratic model $\hat{f}_Q(X) = \hat{\beta}_0 + \hat{\beta}_1 X + \hat{\beta}_2 X^2$ fits slightly better.

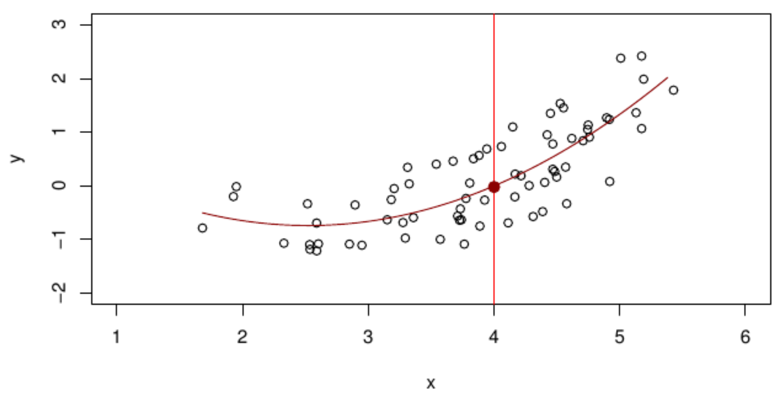



Figure 4: Examples of Parametric Models

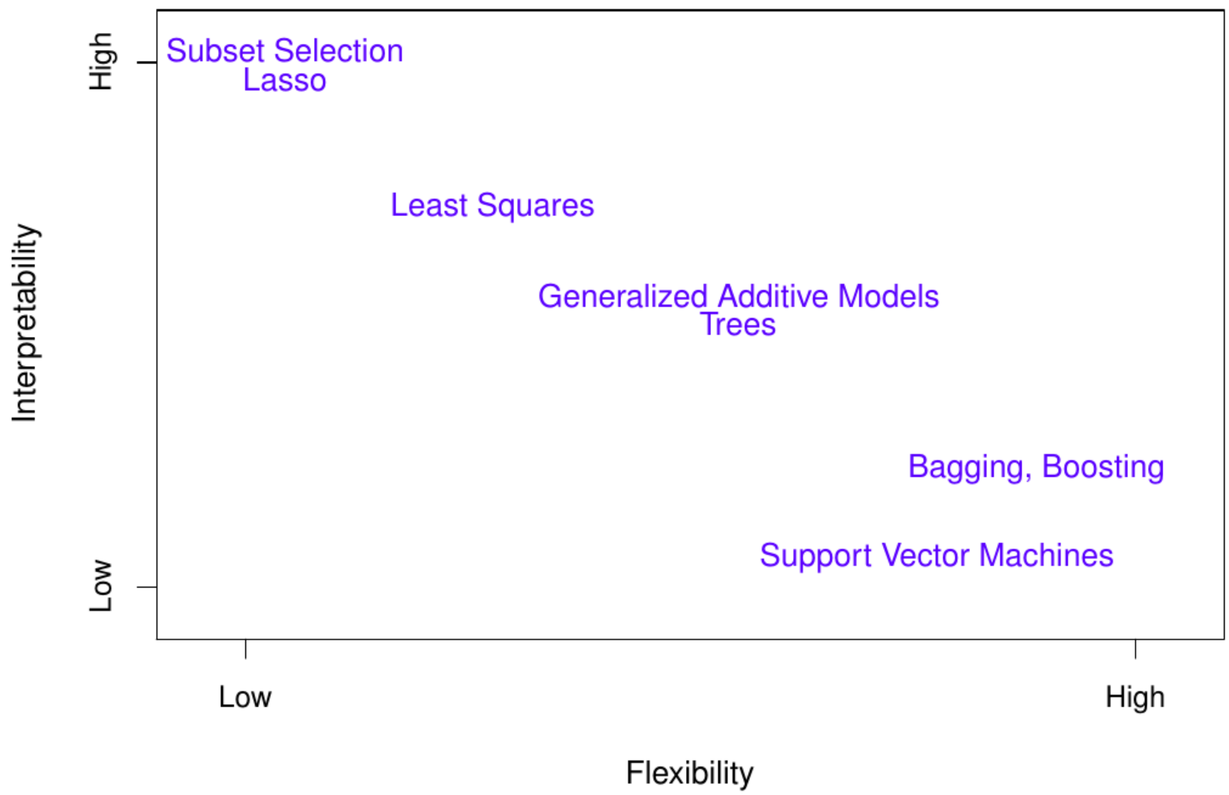



Figure 5: Interpretability vs Flexibility

#### 10.2.1.7.2 Interpretability vs Flexibility

Here are some of the approaches we'll look at this semester

- Simpler models could be more interpretable
  - Or could be too naive
- Flexibility makes for good fits
  - But can lead to overfitting

### 10.2.1.8 Assessing Model Accuracy

#### 10.2.1.8.1 Have to use training (Tr) and testing (Te) datasets

To determine the best predictive model

### 10.2.1.9 The Bias vs. Variance Trade-off

- The hat is the estimated value of something. $\hat{f}(X)$
- We can see the variance of $\hat{f}(X)$
- And the bias in $\hat{f}(X)$

Choosing the flexibility of your fitting function

- (i.e the number of predictors, or coefficients, in your model function)
- based on average test error
- amounts to what we call a bias-variance trade-off

## Assessing Model Accuracy

Suppose we fit a model $\hat{f}(x)$ to some training data $\mathsf{Tr} = \{x_i, y_i\}_1^N$, and we wish to see how well it performs.

- We could compute the average squared prediction error over $\mathsf{Tr}$:

$$\text{MSE}_{\mathsf{Tr}} = \text{Ave}_{i \in \mathsf{Tr}}[y_i - \hat{f}(x_i)]^2$$

This may be biased toward more overfit models.

- Instead we should, if possible, compute it using fresh *test* data $\mathsf{Te} = \{x_i, y_i\}_1^M$:

$$\text{MSE}_{\mathsf{Te}} = \text{Ave}_{i \in \mathsf{Te}}[y_i - \hat{f}(x_i)]^2$$

Figure 6: Assessing Model Accuracy

## Bias-Variance Trade-off

Suppose we have fit a model $\hat{f}(x)$ to some training data $\mathsf{Tr}$, and let $(x_0, y_0)$ be a test observation drawn from the population. If the true model is $Y = f(X) + \epsilon$ (with $f(x) = E(Y|X = x)$), then

$$E\left(y_0 - \hat{f}(x_0)\right)^2 = \mathrm{Var}(\hat{f}(x_0)) + [\mathrm{Bias}(\hat{f}(x_0))]^2 + \mathrm{Var}(\epsilon).$$

The expectation averages over the variability of $y_0$ as well as the variability in $\mathsf{Tr}$. Note that $\mathrm{Bias}(\hat{f}(x_0))] = E[\hat{f}(x_0)] - f(x_0)$.

Typically as the *flexibility* of $\hat{f}$ increases, its variance increases, and its bias decreases. So choosing the flexibility based on average test error amounts to a *bias-variance trade-off.*

Figure 7: Bias vs. Variance Trade-off

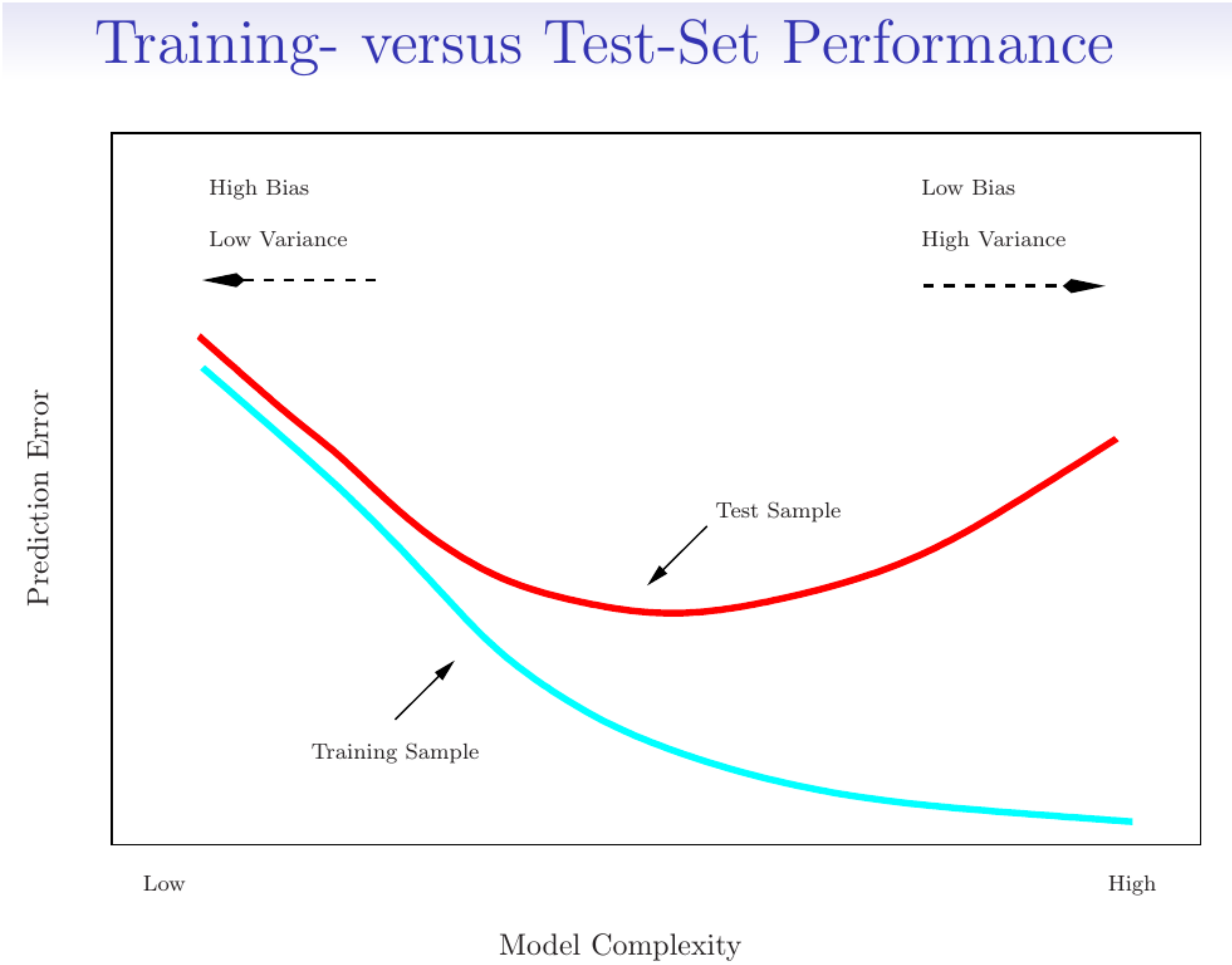



Figure 8: Bias vs. Variance in a Training & Testing Framework

And we use training datasets and testing datasets

- which we apply our model to
- to determine the optimal tradeoff we should use
- for a specific problem and model

#### 10.2.1.9.1 How does all this play out in Classification Problems

As opposed to Regression Problems, which we just discussed

### 10.2.1.10 Citations


- R Core Team. R: A Language and Environment for Statistical Computing. Vienna, Austria: R Foundation for Statistical Computing, 2014..
- Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. An Introduction to Statistical Learning: With Applications in R. 1st ed. 2013, Corr. 5th printing 2015 edition. Springer Texts in Statistics. New York: Springer, 2013.
- Abbass Al Sharif. "Applied Modern Statistical Learning Techniques." [Abbass-Al-Sharif. Accessed January 17, 2016.(http://www.alsharif.info/).

- Diez, David M., Christopher D. Barr, and Mine Çetinkaya-Rundel. OpenIntro Statistics: Third Edition. 3 edition. S.l.: OpenIntro, Inc., 2015.
- Mayor, Eric. Learning Predictive Analytics with R. Packt Publishing - ebooks, 2015.