EECS E6690: Statistical Learning for Biological and Information Systems Lecture1: Introduction

Prof. Predrag R. Jelenković

Time: Tuesday 4:10-6:40pm
303 Seeley W. Mudd Building

Dept. of Electrical Engineering
Columbia University , NY 10027, USA
Office: 812 Schapiro Research Bldg.
Phone: (212) 854-8174
Email: predrag@ee.columbia.edu
URL: http://www.ee.columbia.edu/∼predrag

E6690 Statistical Learning: Brief Description

- **Deluge of Data in Biology and Information Systems**: Ongoing advancements in information systems as well as the emerging revolution in microbiology and neuroscience are creating a deluge of data, whose mining, inference and prediction will have an enormous economic, social, scientific and medical/therapeutic impact.
- **Biology**: For example, in biology, microarray technology is creating vast amounts of gene expression data, whose understanding could lead to better diagnostics and cures of diseases, e.g., cancer. Personalized medicine: designing treatments for individual patients.
- **Information Systems**: Similarly, in information systems, companies like Google, Amazon, Facebook, etc., are facing various problems on massive data sets, e.g., ranking, association and community detection.

E6690 Statistical Learning: Brief Description

This course will cover a variety of fundamental statistical (machine) learning techniques that are suitable for the emerging problems in these application areas, but also applicable in general

- Basics of Statistics and Optimization
- Introduction to Statistical/Machine Leraning Techniques
    - Supervised versus unsupervised learning
    - Inference and prediction
    - Linear versus nonlinear models
    - Training, testing and validation
    - Regularization
    - And many more
- Specifics of Biological and Information Systems Data
    - High dimensionality and need for regularization
    - Large sparse graphs
    - Community detection
    - Ranking
    - Association rules (Market basket analysis)

E6690 Statistical Learning: Course Logistics

Prerequisites: Calculus. Some knowledge of probability/statistics and optimization is strongly encouraged, but not required. Familiarity with a programming language, say Matlab, is highly desirable.

Textbooks: The following two books will represent the supporting references for the course. The books are available online:

ESL Hastie, T., Tibshirani, R. and Friedman, J. The Elements of Statistical Learning: Data Mining, Inference and Prediction, 2nd Edition. Springer, 2009. `https://hastie.su.domains/Papers/ESLII.pdf`.

ISL James, G., Witten, D. Hastie, T. and Tibshirani, R. An Introduction to Statistical Learning, Springer, 2014. Available online at `https://www.statlearning.com`. R code can be found at: `https://hastie.su.domains/ISLR2/Labs/`

In addition, lecture notes as well as occasionally other books and research papers will be used.

Homework: Biweekly homework will be assigned (about 3-4)

Programming: The course uses R language. Pointers to its free download and resources, as well as basic examples of programming in R will be covered in class.

Grading: Homework (20%) + Midterm (35%) + Final Proj (45%).

E6690 Statistical Learning: Course Logistics

Midterm: In class, closed book; 2 page cheat-sheet allowed; 2 1/2 hours

- Mixture of problem solving and descriptive answers
  (This might change since the course is online.)

Final Project: Done in groups of 3 (maybe 4) students

- First, select a paper(s) from a data repository, e.g.:
  - GEO (Gene Expression Omnibus) Data Repository
    `https://www.ncbi.nlm.nih.gov/geo/`
  - UC Irvine Machine Learning Repository
    `https://archive.ics.uci.edu/ml/datasets.php`
- General Project Outline
  1. Introduction: e.g., describe the application area, problems considered, etc
  2. Data set(s) and paper(s): e.g., describe data in detail, what was done in the paper(s), common stat/machine learning tools, etc
  3. Reproduce the results from the paper(s)
  4. Try different techniques learned in class, or propose new ones
  5. Discussion and conclusion: e.g., compare different techniques, pros and cons, future work, etc

Statistical Learning: What Does It Involve?

In general, Statistical (Machine) Learning (supervised) problems typically can be posed as

$$\boxed{Y = f(X) + \epsilon}$$

where ϵ is the nose.

Problem: Estimate f from training data $\{(x_i, y_i)\}$, and then use it as a general solution. Typically: $y \in \mathbb{R}$: regression; or y-discrete: classification

Two main setups:

- Noiseless case ($Y = f(X)$): more common in machine learning
- Noisy case ($Y = f(X) + \epsilon$): more prevalent in statistics

Areas involved:

- Approximation theory - for picking a class of functions
- Optimization - for fitting the training data
- Computing - fitting and testing
- Probability and Statistics - testing, error estimation

Machine Learning Versus Classical Programming

Interesting Question: What is the difference between classical programming and statistical/machine learning?

$$\boxed{Y = f(X)}$$

- Classical Programming: f is an algorithm designed by a person
- Statistical Learning: f is discovered through examples by training

General Course Objectives

- Focus/motivation - emerging applications in:
  - Biology and Medicine
  - Information Technology, e.g., problems arising from: Google, Facebook, Twitter, Amazon, etc.
- Learn fundamental concepts and techniques in statistical (machine) learning techniques that are
  - Suitable for these application areas
  - Useful and applicable in general
- Develop the necessary knowledge as we go (e.g., Statistics, Optimization, Approximation Theory, etc)
- Learn R
- Have a hands-on experience on a real, practical problem through a final project

Overall objective: Become an expert in Statistical/Machine Learning

Programming in R: Computing Platform

- Language and environment for statistical computing and graphics
- Free software
- Download
  - R from `http://cran.r-project.org/`
  - RStudio, an Integrated Development Environment for R, from `http://www.rstudio.com/products/rstudio/download/`
- Resources
  - R for beginners
  - Quick-R
  - Cookbook for R
  - R for Data Science
  - Try R

Brief Statistics Review

Crash Course in Undergraduate Statistics

Example

The following numbers are particle (contamination) counts for a sample of 10 semiconductor silicon wafers:

50 48 44 56 61 52 53 55 67 51

Over a long run the process average for wafer particle counts has been 50 counts per wafer, and on the basis of the sample, we want to test whether a change has occurred.

Is data consistent with a given hypothesis?

- Idea: Data $\rightarrow$ estimate with a known distribution
  (Estimates are not unique)
- Is the estimate consistent the hypothesized distribution?
  How likely is the estimate?

Estimates

- A statistic is a property of sample data taken from a population
- A point estimate of some unknown parameter is a statistic that provides a best guess at the parameter value
- A point estimate $\hat{\theta}$ is **unbiased** if $\mathbb{E}\hat{\theta} = \theta$
- $X_1, X_2, \ldots, X_n$ – i.i.d. with population mean μ & variance σ^2
- Examples
  - Sample mean - estimate of the population mean μ
  $$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$$
  - Sample variance - estimate of the population variance σ^2
  $$S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2$$
  - Variability: $\mathsf{Var}(\bar{X}) = \sigma^2/n$, but σ unknown
  replace σ^2/n with *standard error (SE)*, $\mathsf{SE}(\bar{X})^2 := S^2/n$
  $\Rightarrow \quad \mathsf{Var}(\bar{X}) = \sigma^2/n \approx \mathsf{SE}(\bar{X})^2$

Variability of estimates: Known variance

- If $X_1, \dots, X_n$ are **i.i.d. normal**, then
  - $\bar{X}$ is normal:
  $$\frac{\bar{X} - \mu}{\sqrt{\sigma^2/n}} \sim \mathcal{N}(0,1)$$
  - S^2 has a known distribution:
  $$\frac{n-1}{\sigma^2} S^2 \sim \chi^2_{n-1},$$
  where χ^2_{n-1} (Chi - square) is a random variable whose distribution is equal to the sum of $(n-1)$ squares of independent standard normal random variables
  - $\bar{X}$ and S^2 are independent (prove)
- If $X_1, \dots, X_n$ are **not** i.i.d normal, then CLT:
$$\frac{\bar{X} - \mu}{\sqrt{\sigma^2/n}} \Rightarrow \mathcal{N}(0,1)$$

Variability of estimates: Unknown variance

- If $X_1, \ldots, X_n$ are **i.i.d. normal**, then
  - t-statistic:

$$\frac{\bar{X} - \mu}{\sqrt{S^2/n}} \sim \frac{\mathcal{N}(0,1)}{\sqrt{\chi^2_{n-1}/(n-1)}} \sim t_{n-1},$$

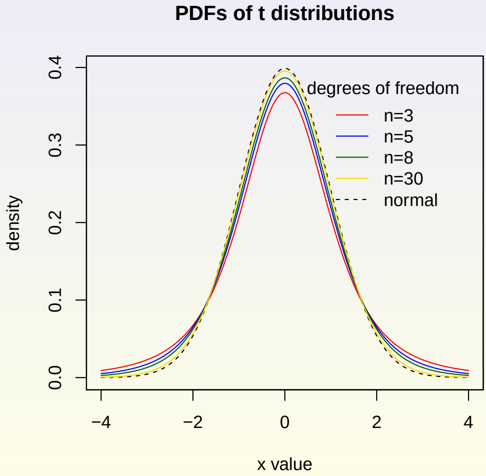
    t_{n-1} is Student's t-distribution with $(n-1)$ degrees of freedom
    Developed by William Gosset; worked for Guinness brewery, published the paper under the pen name Student

- Representation of t_n: Let $Z \sim \mathcal{N}(0,1)$ and $V \sim \chi^2_n$ be independent

$$\frac{Z}{\sqrt{V/n}} \sim t_n$$

t-distribution

- Zero mean
- Variance $(n > 2)$: $n/(n-2)$



t-test

- Null hypothesis $\mathcal{H}_0 : \mu = \mu_0$
- Under $\mathcal{H}_0$, t-statistic:

$$t = \frac{\bar{X} - \mu_0}{\sqrt{S^2/n}} \sim t_{n-1}$$

  and the corresponding p-value is the probability of observing $|t_{n-1}|$ that is $\geq |t|$, i.e., $p = \mathbb{P}[|t_{n-1}| \geq |t|]$.
- Large values of t unlikely under $\mathcal{H}_0$
- Typically:
  - pick a significance value, say $\alpha = 0.05$ (not unique)
  - reject if $p < \alpha$, say $p < 0.05$
  - accept if $p \geq \alpha$, say $p \geq 0.05$

Intro to Statistical Learning

Supervised vs. unsupervised learning

- Supervised learning: there is an input-output relationship

$$\boxed{Y = f(X) + \epsilon}$$

  - $X \in \mathbb{R}^p$ - Vector of p predictor measurements
  - $Y \in \mathbb{R}$ - Outcome measurements
  - ϵ: noise
  - Two problems:
    - Regression: Y is quantitative/real
    - Classification: Y is categorical/discrete
  - Training data (observations): $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$
  - Objectives:
    - Statistics: Prediction, inference
    - Machine learning: Solve a problem via training
- Unsupervised learning: No outcome variable Y
  - Objective can be vague - just exploring data
  - Learn interesting phenomena in data, e.g.:
    - Clustering, community detection, data association, low dimensional representation

Learning

- Let $Y \in \mathbb{R}$ be the output variable, and $X \in \mathbb{R}^p$ the input vector $X = (X_1, X_2, \ldots, X_p)$. Then

$$Y = f(X) + \epsilon$$

- Want to estimate what f is
- ϵ is unavoidable noise that is independent of X, zero mean
- How to estimate f from the data? How to evaluate the estimate?
- Given an estimate $\hat{f}$ for f, predict unavailable values of Y for known values of X: $\hat{Y} = \hat{f}(X)$
- Reducible and irreducible errors:
  - $\hat{f}$ is not exactly f, but f can potentially be learnt given enough data
  - even if f is known, there is error: $\epsilon = Y - f(X)$

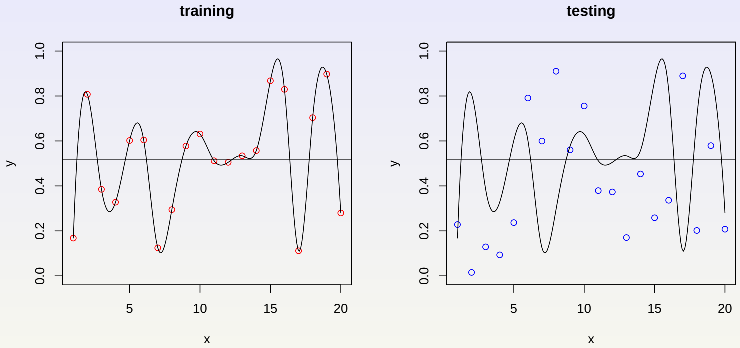
Two approaches to estimate f

- Parametric
  - Assume a specific form of f
  - Example: the linear model

$$\hat{f}(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p$$

  - Use training data to choose the values of parameters $\beta_0, \beta_1, ..., \beta_p$
  - Pro: easier to estimate parameters than arbitrary function
  - Con: the choice of f might be (very) wrong
- Non-parametric
  - Make the parametric form more flexible
  - This makes $\hat{f}$ more complex and potentially following the noise too closely, thereby **overfitting**
  - Get f as close as possible to the data points, subject to not being too non-smooth
  - Pro: more likely to get f right, especially if f is “strange”
  - Con: more data is needed to obtain a good estimate for f

Example



- More complicated models not always better - e.g., overfitting
  John von Neumann: "With four parameters I can fit an elephant, and with five I can make him wiggle his trunk."
  (Deep learning: 50,000,000 parameters (!))
- Amount of available data
- Interpretability

Linear Regression

Idea

- Simple approach to supervised learning
- Assumes linear dependence of quantitative Y on $X_1, X_2, \ldots, X_p$
- True regression functions are never linear!
  - But most learning methods linear in parameters (β-s)!
- Extremely useful both conceptually and practically

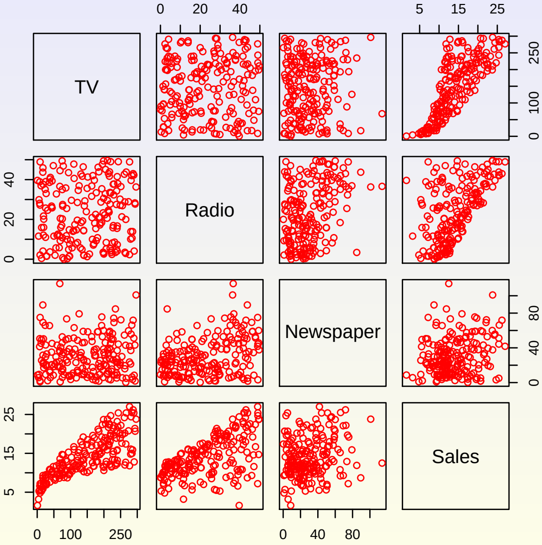
Data set

- Will use `Advertising.csv` to illustrate concepts
- 200 observations:

```
"","TV","Radio","Newspaper","Sales"
"1",230.1,37.8,69.2,22.1
"2",44.5,39.3,45.1,10.4
"3",17.2,45.9,69.3,9.3
.
.
.
"198",177,9.3,6.4,12.8
"199",283.6,42,66.2,25.5
"200",232.1,8.6,8.7,13.4
```

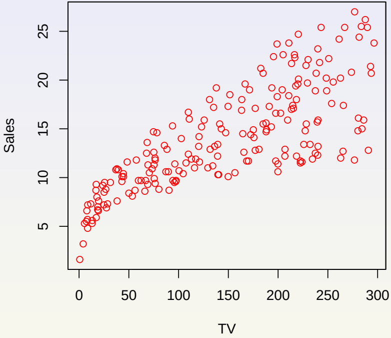
Advertising data set

Visualize data - whenever possible



Single predictor: TV vs. Sales

```
> adv<-read.csv("advertising.csv",header=TRUE,sep=",")
> plot(adv$TV,adv$Sales,xlab="TV",ylab="Sales",col="red")
```



- Linear model
$$Y = \beta_0 + \beta_1 X + \epsilon,$$
where
  - β_0 and β_1: unknown constants/parameters/coefficients (intercept and slope)
  - ϵ: error term

Single predictor: Model selection

- Estimate β_0 and β_1 based on data
- Given estimates $\hat{\beta}_0$ and $\hat{\beta}_1$, predict future sales using

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$$

- $\hat{y}$: prediction of Y given $X = x$
- **Residuals**: $y_i - \hat{y}_i = y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i)$

- Select $\hat{\beta}_0$ and $\hat{\beta}_1$ to "minimize" residuals

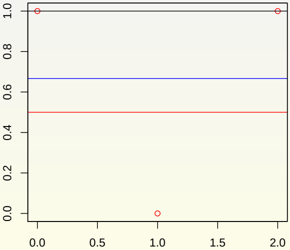
- How to minimize a vector?

Need to Define Distance: Vector norms

- Example: l_p norm

$$\|\boldsymbol{z}\|_p = \left(\sum_{i=1}^{n} |z_i|^p \right)^{1/p}$$

- Example: 3 data point - $\{(0,1),(1,0),(2,1)\}$
  The result depends on the choice of the norm (!)
  (parallel to x-axis due to symmetry)



One dimensional l_2 regression: Least squares

- $\min \|\boldsymbol{y} - \hat{\boldsymbol{y}}\|_2$
- Residual Sum of Squares (RSS):

$$\mathsf{RSS} \equiv \mathsf{RSS}(\beta_0, \beta_1) = \|\boldsymbol{y} - \hat{\boldsymbol{y}}\|_2^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

- Least squares approach: $\min_{\beta_0, \beta_1} \mathsf{RSS}$
- Solution:

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} (y_i - \bar{y})(x_i - \bar{x})}{\sum_{i=1}^{n} (x_i - \bar{x})^2},$$
$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x},$$

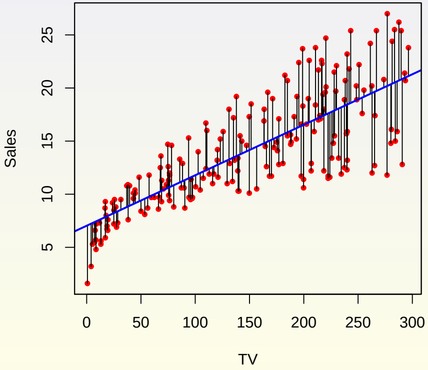
where $\bar{x} = n^{-1} \sum_{i=1}^{n} x_i$ and $\bar{y} = n^{-1} \sum_{i=1}^{n} y_i$ are the sample means

Example

```
> lm1<-lm(adv$Sales~adv$TV)
> summary(lm1)
```

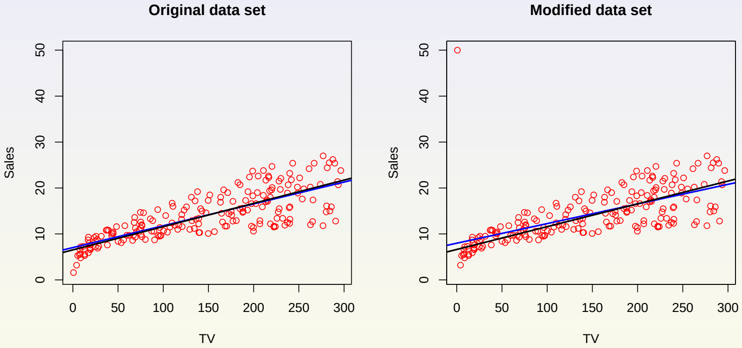
$$\text{Sales} = 7.032594 + 0.047537 \times \text{TV}$$

```
> plot(adv$TV,adv$Sales,xlab="TV",ylab="Sales",col="red",pch=20)
> abline(lm(adv$Sales~adv$TV),col="blue",lwd=2)
> Sales_Predict<-predict(lm1)
> segments(adv$TV, adv$Sales, adv$TV, Sales_Predict)
```



Example: l_2 vs. l_1

- One point in the data set modified

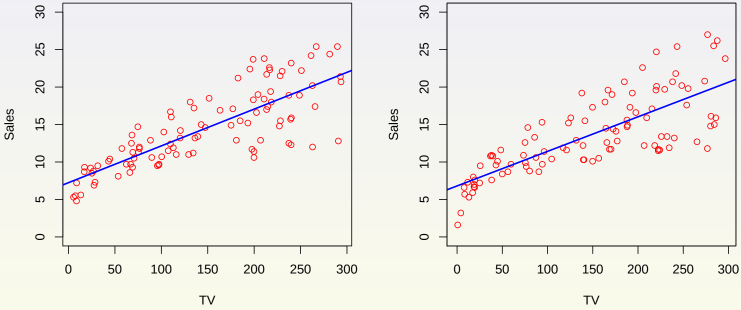


Coefficient estimates

- Suppose the true model is

$$\text{Sales} = \beta_0 + \beta_1 \times \text{TV} + \epsilon$$

- How good are estimates $\hat{\beta}_0$ and $\hat{\beta}_1$?



$$i = 1, \ldots, 100: \quad \text{Sales} = 7.241734 + 0.049069 \times \text{TV}$$
$$i = 101, \ldots, 200: \quad \text{Sales} = 6.803818 + 0.046135 \times \text{TV}$$

Properties of $\hat{\beta}_0$ and $\hat{\beta}_1$

- Repeated sampling
- $\hat{\beta}_0$ and $\hat{\beta}_1$ vary
- Means:

$$\mathbb{E}\hat{\beta}_0 = \beta_0 \quad \text{and} \quad \mathbb{E}\hat{\beta}_1 = \beta_1$$

- Variances:

$$\mathrm{Var}(\hat{\beta}_1) = \frac{\sigma^2}{\sum_{i=1}^n (x_i - \bar{x})^2},$$

$$\mathrm{Var}(\hat{\beta}_0) = \sigma^2 \left(\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^n (x_i - \bar{x})^2} \right),$$

where $\sigma^2 = \mathrm{Var}(\epsilon)$

- An estimate of σ^2:

$$\mathrm{RSE}^2 = \frac{1}{n-2} \sum_{i=1}^n (y_i - \hat{y}_i)^2 = \frac{1}{n-2}\mathrm{RSS},$$

where RSE is the Residual Standard Error

Confidence intervals

- Normality assumption: $\epsilon \sim \mathcal{N}(0, \sigma^2)$
- t-statistic:

$$\frac{\hat{\beta}_1 - \beta_1}{\mathsf{SE}(\hat{\beta}_1)} \sim t_{n-2},$$

  where

$$\mathsf{SE}(\hat{\beta}_1)^2 = \frac{1}{n-2} \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (x_i - \bar{x})^2}$$

- $(1-\gamma)$ confidence interval (example $\gamma = 5\%, 1-\gamma = 95\%$)

$$[\hat{\beta}_1 - \mathsf{SE}(\hat{\beta}_1) \cdot t_{\gamma/2,n-2}, \hat{\beta}_1 + \mathsf{SE}(\hat{\beta}_1) \cdot t_{\gamma/2,n-2}]$$

  is such that

$$\mathbb{P}[\beta_1 \in [\hat{\beta}_1 - \mathsf{SE}(\hat{\beta}_1) \cdot t_{\gamma/2,n-2}, \hat{\beta}_1 + \mathsf{SE}(\hat{\beta}_1) \cdot t_{\gamma/2,n-2}]] = 1 - \gamma,$$

  where $t_{\gamma/2,n-2}$ is the $(1-\gamma/2)$-th quantile of the t_{n-2} distribution

Hypothesis testing

- Typical testing (null vs. alternative hypothesis):

  $\mathcal{H}_0$: there is no relationship between X and Y
  versus alternative
  $\mathcal{H}_A$: there is some relationship between X and Y

- Formally:

$$\mathcal{H}_0 : \beta_1 = 0 \quad \text{vs.} \quad \mathcal{H}_A : \beta_1 \neq 0$$

- To test $\mathcal{H}_0$ ($\beta_1 = 0$), compute a t-statistic:

$$t = \frac{\hat{\beta}_1 - 0}{\mathsf{SE}(\hat{\beta}_1)},$$

  which is distributed according to a t-distribution with $(n-2)$ degrees of freedom

- Compute the p-value – probability of observing any value equal to $|t|$ or larger

Example

```
> summary(lm1)

Call:
lm(formula = adv$Sales ~ adv$TV)

Residuals:
    Min      1Q  Median      3Q     Max
-8.3860 -1.9545 -0.1913  2.0671  7.2124

Coefficients:
            Estimate Std. Error  t value  Pr(>|t|)
(Intercept) 7.032594   0.457843  15.36   <2e-16 ***
adv$TV      0.047537   0.002691  17.67   <2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 3.259 on 198 degrees of freedom
Multiple R-squared:  0.6119,       Adjusted R-squared:  0.6099
F-statistic: 312.1 on 1 and 198 DF,  p-value: < 2.2e-16


> qt(0.975,198)
[1] 1.972017
```

Reading:

ISL: Read in detail Chapter 2 and Section 3.1.
Also, looking through the entire Chapters 1-3 is recommended.