



W261 - Week 6

**Distributed Supervised Machine Learning:
Part I
Linear Regression**

UC Berkeley - MIDS



Table of Contents

- Housekeeping
- Intro to linear regression
 - MSE, Lambda functions
 - Gradient descent
 - Linear regression at scale in Spark

MidTerm Course Evaluation (due this week)

Please fill out the course evals over the next 24 hours (if possible) here. These are important for the course and for the program. Thank you!!

<https://course-evaluations.berkeley.edu/berkeley/>

Thank you!

Take 10
minutes to
complete now
(due this week)



Customer Satisfaction Survey

[Title of Project]

Goal: Explain what you want to get out of conducting this customer survey.

Timeline: What is your timeline for completion?

Team: List out the team members involved with this project.

Directions:

Hi Team,

Let's collaborate on our [project name] customer satisfaction survey. We need to finalize our list of survey questions, create a survey form (we can use either Google Docs or Typeform). Let's track the results via Google Spreadsheet.

Your Team Leader

Step 1: Finalize Survey Questionnaire

1) How likely are you to recommend this company to a friend or colleague? (NPS)

• Scale 0-10 (not likely to extremely likely)

2) How satisfied or dissatisfied are you with our company?

Week 6

Course Outline

[Live Session Plan](#)[Study List](#)[Wall](#)[Toolbox](#)[Course Overview](#)

Assessments

[1: Introduction and Motivation for M...](#)[2: Parallel Computation Frameworks](#)[3: Map-Reduce Algorithm Design](#)[4: Introduction to Spark With RDDs: ...](#)[5: Introduction to Spark With RDDs: ...](#)**[6: Distributed Supervised Machine L...](#)**[7: Distributed Supervised Machine L...](#)[8: Data Systems and Pipelines](#)[9: Graph Algorithms at Scale](#)[10: Large Scale Graph Processing: R...](#)[11: Distributed Decision Trees for Cla...](#)[12: Mid-Project Presentations \(Optio...](#)[13: Alternating Least Squares, Vario...](#)[14: Final Project Presentations](#)[Live Sessions](#)

6: Distributed Supervised Machine Learning: Part I



 1h 36m Total Video Time

Course Content


Description

6.1 Weekly Introduction 6	Sequence	2m 15s	View Sequence Outline
6.2 Supervised Machine Learning Parametric vs. Nonsupervised	Interactive Video	8m 18s	
6.3 Core Supporting Concepts: Matrices, Applications of Matrices	Interactive Video	4m 46s	
6.4 Matrices: Vector-by-Vector Multiplication	Interactive Video	11m 33s	
6.5 Quiz 6-1	Sequence		View Sequence Outline
6.6 Matrices: Matrix-by-Matrix Multiplication, Version 1.0	Interactive Video	12m 3s	
6.7 Matrices: Distributed Matrix-by-Vector Multiplication, Version 1.1	Interactive Video	5m 58s	
6.8 Distributed Matrix Summary	Interactive Video	3m 2s	
6.9 Core Supporting Concepts: Optimization Theory—Gradient Descent, Part 1	Interactive Video	9m 14s	
6.10 Core Supporting Concepts: Optimization Theory—Gradient Descent, Part 2	Interactive Video	8m 57s	
6.11 Optimization Theory: Convex Optimization	Interactive Video	11m 29s	
6.12 Distributed Closed-Form Linear Regression	Interactive Video	4m 46s	
6.13 Quiz 6-2	Sequence		View Sequence Outline
6.14 Complexity Analysis of Algorithms	Interactive Video	3m 28s	
6.15 Distributed-Gradient-Descent Approach to Linear Regression	Interactive Video	7m 21s	
6.16 Quiz 6-3	Sequence		View Sequence Outline


HW3




Account




Dashboard




Courses




Calendar




Inbox




History




Commons



Help

 w261 > Assignments

Home

Announcements 

Assignments

Discussions


Grades

People

Pages

Files

Syllabus

Outcomes 


Rubrics


Quizzes


Modules


BigBlueButton


Collaborations


 Assignments

 **HW1 - Intro to the Map Reduce Paradigm**
Module 1: introduction to machine learning at scale Module | Due May 15 at 11:59pm | 100 pts

 **HW2 - Naive Bayes in Hadoop MR**
Module 2: Hadoop Map-Reduce basics Module | Due May 29 at 11:59pm | 100 pts

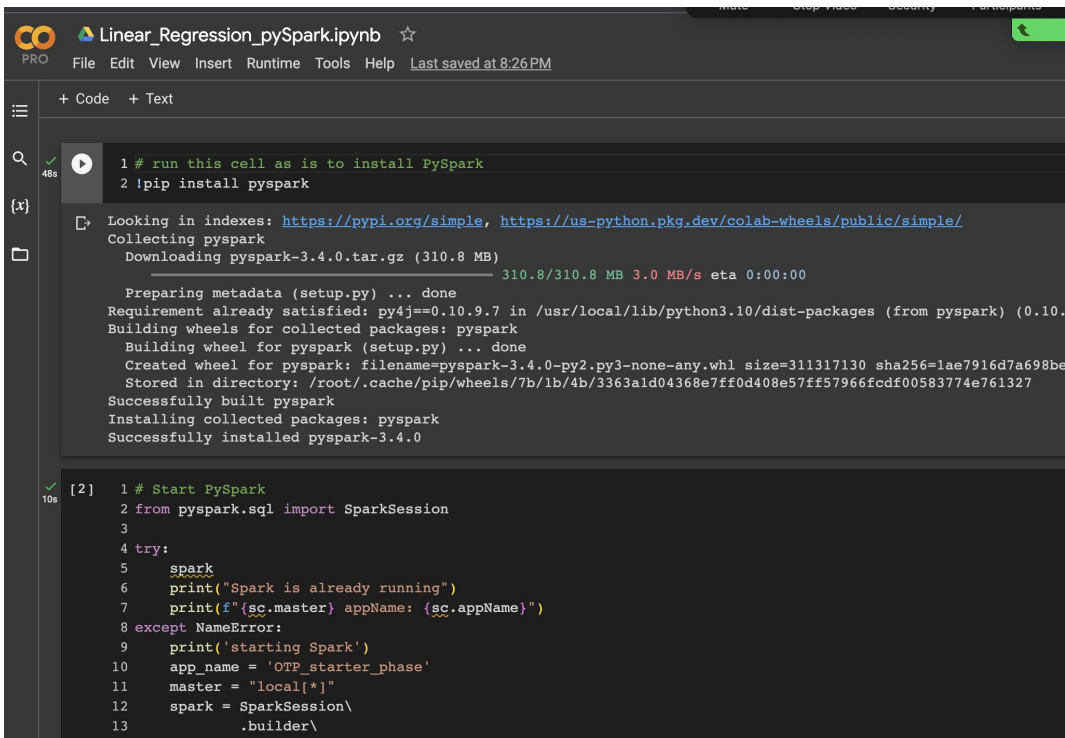
 **HW3 - Synonym Detection using Apache Spark**
Module 4: Introduction to Apache Spark Map-Reduce (Part 1 of 2) Module | Due Jun 15 at 11:59pm | 100 pts

 Imported Assignments

 **Module 5 Homework: PyTorch Basics**
Due Jul 27, 2021 at 11:59pm | 30 pts

The following is a link to a colab notebook for week06's live session (partial notebook):

https://colab.research.google.com/drive/1Dpp_puKb3XnmRtOfDJlbuSEfubKZPrY?usp=sharing



```
Linear_Regression_pySpark.ipynb ☆
PRO File Edit View Insert Runtime Tools Help Last saved at 8:26 PM

+ Code + Text

1 # run this cell as is to install PySpark
2 !pip install pyspark

Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting pyspark
  Downloading pyspark-3.4.0.tar.gz (310.8 MB)
    310.8/310.8 MB 3.0 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: py4j==0.10.9.7 in /usr/local/lib/python3.10/dist-packages (from pyspark) (0.10.9.7)
Building wheels for collected packages: pyspark
  Building wheel for pyspark (setup.py) ... done
  Created wheel for pyspark: filename=pyspark-3.4.0-py2.py3-none-any.whl size=311317130 sha256=1ae7916d7a698be8e8e8e8e8e8e8e8e8e8e8e8e8e8e8e8e8e8e8e8e8e8e8e8e8e
  Stored in directory: /root/.cache/pip/wheels/7b/1b/4b/3363a1d04368e7ff0d408e57ff57966fcd00583774e761327
Successfully built pyspark
Installing collected packages: pyspark
Successfully installed pyspark-3.4.0

[2] 1 # Start PySpark
2 from pyspark.sql import SparkSession
3
4 try:
5     spark
6     print("Spark is already running")
7     print(f'{sc.master} appName: {sc.appName}')
8 except NameError:
9     print('starting Spark')
10    app_name = 'OTP_starter_phase'
11    master = "local[*]"
12    spark = SparkSession\
13        .builder\
```

DEMO6_WORKBOOK.IPYNB

3.1.0.1. A warning about OLS before we start:

3.2. Notebook Set Up

4. A Small Example

4.0.0.1. lambda functions

4.0.1. Calculate the MSE in Spark

4.0.1.1. Non-augmented data

4.0.1.2. Calculate Gradient using Numpy on single core

4.0.1.3. Calculate Gradient using spark in local model on my laptop

4.1. Linear Regression

4.2. Ridge Linear Regression

4.3. Lasso Linear Regression

4.4. Elastic Net Linear Regression

4.5. Demo: Random Parameter Search.

4.6. Demo: Systematic Brute Force.

5. Parameter Space, Gradients, and Convexity

5.0.1. Optimization Theory ... a short digression

5.1. Demo: Gradient Descent

5.2. Demo : Stochastic Gradient Descent

5.2.1. That's it for today!

demo4_workbook.ipynb x demo5_workbook-KMear x demo6_workbook.ipynb x hw2_workbook.ipynb x hw3_Workbook.ipynb x demo6_workbook.ipynb x

Python 3

```
[13]: 24
```

```
[24]: 1 model = np.array([0, 1])
      2 (lambda X_y: print(f"y_pred: {np.dot(model, X_y[: -1])}"))([1,1,2])
      3
```

y_pred: 1

```
[21]: 1 model = np.array([0, 1]) #augmented model
      2 (lambda X_y: print(f"error: {np.dot(model, c) - X_y[-1]}"))([1, 1, 2])
      3
```

error: -1

```
[22]: 1 model = np.array([0, 1]) #augmented model
      2 (lambda X_y: print(f"squared error: {(np.dot(model, X_y[: -1]) - X_y[-1])**2}"))([1, 1, 2])
      3
```

squared error: 1

4.0.1.1. Non-augmented data

```
[29]: 1 model = np.array([0, 1]) #augmented model
      2 #non-augmented data
      3 (lambda X_y: print(f"squared error: {(np.dot(model, np.append(1,X_y[: -1])) - X_y[-1])**2}"))([1, 2])
      4
```

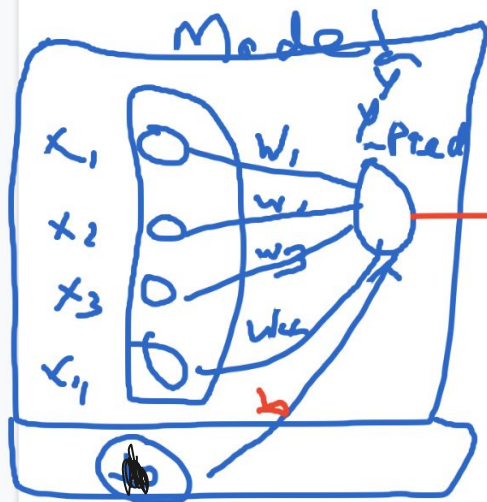
squared error: 1

```
[31]: 1 # raw lambda function
      2 model = np.array([0, 1]) #augmented model
      3 (lambda X_y: (np.dot(model, np.append(1,X_y[: -1])) - X_y[-1])**2)([1, 2])
```

```
[31]: 1
```

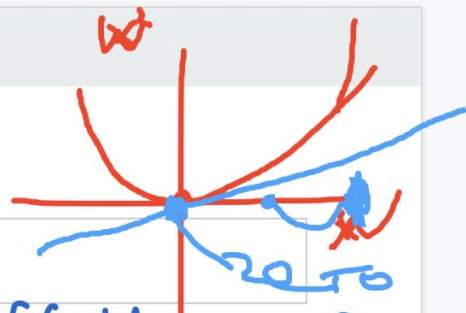
```
[2]: 1 # run this cell once only to start the
      2 import re
      3 import numpy as np
      4 import pandas as pd
      5 import pyspark
      6
      7 # initialize spark context
```

Click to add title



Minimize $f = w^2$

$\frac{\partial f}{\partial w} = 2w$



Learning Scaffolding
Loss

MSE
GD

Minimize MSE = $\frac{1}{n} \sum (x(w) - y)^2$



Quality

root of df

A TASK FOR YOU: Perform 5 steps of Stochastic Gradient Descent (i.e., where the batchsize is 1 on the small data set that was used previously).

Assume that the following setup for gradient descent where:

- an initial weight vector of $\theta = [0 \quad 1]$ corresponding to the y-intercept and slope of the simple linear regression model
- and learning rate of $\eta = 0.1$

In the following table one iteration of SGD has already been calculated for illustration purposes. Now complete the remaining 4 iterations of stochastic gradient descent and fill the following table (note that the code corresponding to these calculations is provide below):

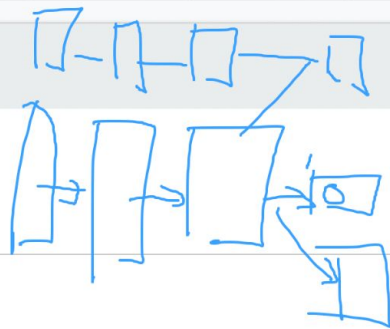
x_j	y_j	$x_j' \cdot \theta$	$\frac{2}{m} [x_j' \cdot \theta - y_j] \cdot x_j'$	$\eta \nabla_{\theta} f$	$(\theta) - \eta \nabla_{\theta} f$
input	true y	predicted y	gradient for this 'batch'	update	new θ
$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$	2	1	$\nabla = \frac{2}{1} [-1] \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix}$	$\nabla = \begin{bmatrix} -0.2 \\ -0.2 \end{bmatrix}$	$\begin{bmatrix} 0.2 \\ 1.2 \end{bmatrix}$
$\begin{bmatrix} 1 \\ 3 \end{bmatrix}$	4	3.8	$\frac{2}{4} [-0.2] \cdot \begin{bmatrix} 1 \\ 3 \end{bmatrix}$	$\begin{bmatrix} -0.04 \\ -0.12 \end{bmatrix}$	$\begin{bmatrix} 0.2 \\ 1.2 \end{bmatrix}$
$\begin{bmatrix} 1 \\ 5 \end{bmatrix}$	5				
$\begin{bmatrix} 1 \\ 4 \end{bmatrix}$	3				
$\begin{bmatrix} 1 \\ 2 \end{bmatrix}$	3				

DISCUSSION QUESTIONS:

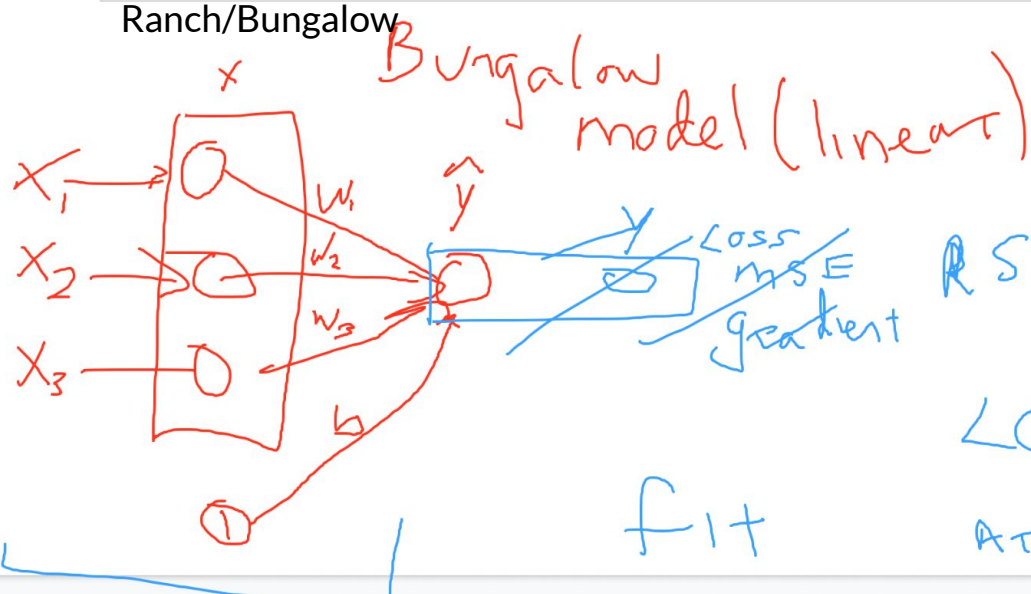
- How does this result compare to our result from the hand calculations in the last section? What implications does this have for our quest to find the

Click to add title

Ranch/Bungalow



x_1	x_2	x_3	y
			0



RSS + L1 + L2

LOSS
+
Architecture

Linear Regression - review

James G., Witten D., Hastie T., Tibshirani R. (2013) Linear Regression. In: An Introduction to Statistical Learning. Springer Texts in Statistics, vol 103. Springer, New York, NY

Chapter 3.1, 3.2, 3.3

3.1 Simple Linear Regression

3.1.1 Estimating the coefficients - Ordinary Least Squares (OLS)

We want to find the line as close as possible to all the data points.

$$Y \approx \beta_0 + \beta_1 X$$

A common way to measure closeness involves minimizing the **least squares** criterion.

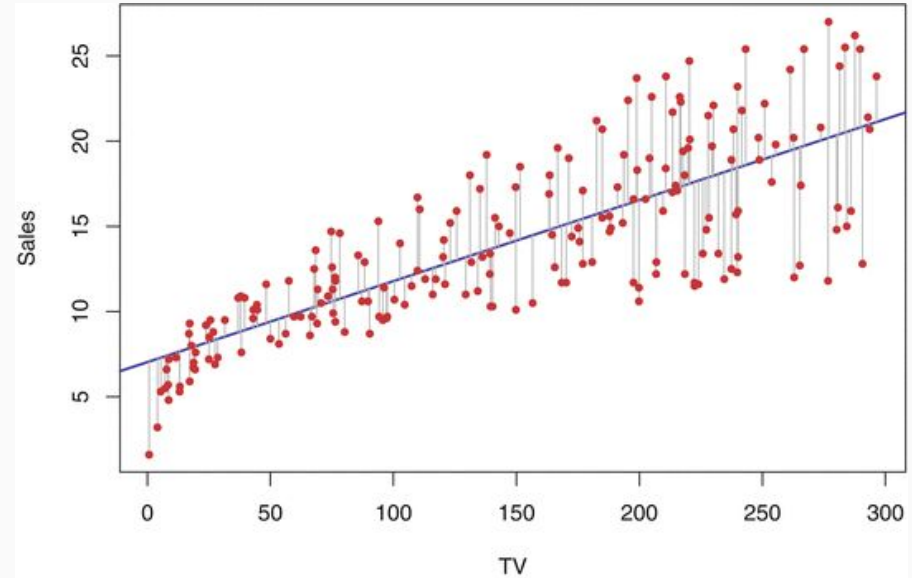


Fig. 3.1

For the Advertising data, the least squares fit for the regression of sales onto TV is shown. The fit is found by minimizing the sum of squared errors. Each grey line segment represents an error, and the fit makes a compromise by averaging their squares. In this case a linear fit captures the essence of the relationship, although it is somewhat deficient in the left of the plot.

3.1.1 Estimating the coefficients

The Least Squares approach chooses $\hat{\beta}_0$ and $\hat{\beta}_1$ to minimize the **Residual Sum of Squares (RSS)**.

$$\text{RSS} = e_1^2 + e_2^2 + \dots + e_n^2$$

where

$$e_i = y_i - \hat{y}_i$$

Residual
error =
observed y - estimate of y

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

Prediction
estimate of y =
estimates of $B_0 + B_1 * x$

$$\text{RSS} = (y_1 - \hat{\beta}_0 - \hat{\beta}_1 x_1)^2 + (y_2 - \hat{\beta}_0 - \hat{\beta}_1 x_2)^2 + \dots + (y_n - \hat{\beta}_0 - \hat{\beta}_1 x_n)^2$$

$$\text{MSE}(\theta; \mathbf{X}, \mathbf{y}) = \frac{1}{m} \sum_{i=1}^m (\mathbf{x}^{(i)} \cdot \theta - y^{(i)})^2$$

Partial derivatives notation :

$$\frac{\partial}{\partial \theta_j} \text{MSE}(\theta)$$

Partial derivatives of the cost function with respect a single input θ_j

$$\frac{\partial}{\partial \theta_j} \text{MSE}(\theta) = \frac{2}{m} \sum_{i=1}^m (\mathbf{x}^{(i)} \cdot \theta - y^{(i)}) x_j^{(i)}$$

Gradient vector of the cost function vectorized

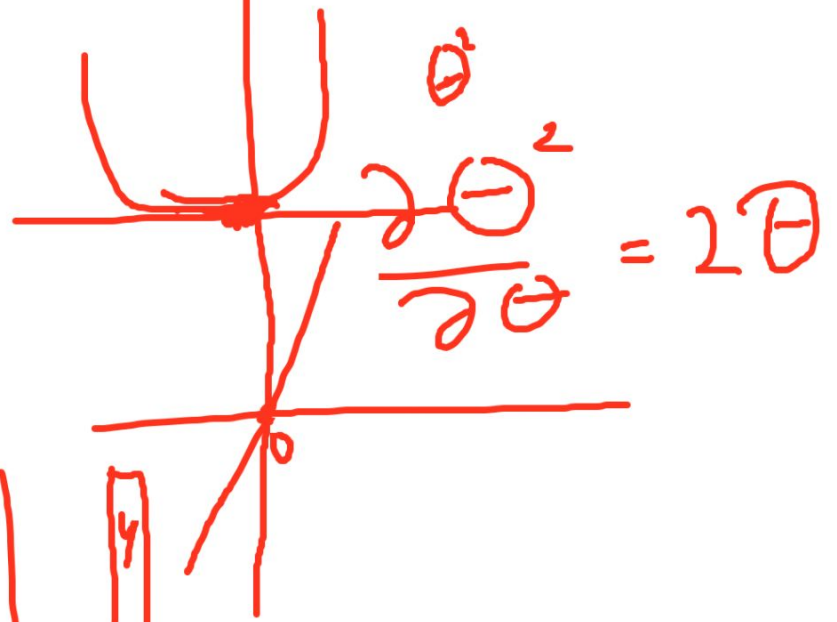
$$\nabla_{\text{MSE}}(\theta) = \begin{pmatrix} \frac{\partial}{\partial \theta_0} \text{MSE}(\theta) \\ \frac{\partial}{\partial \theta_1} \text{MSE}(\theta) \\ \vdots \\ \frac{\partial}{\partial \theta_n} \text{MSE}(\theta) \end{pmatrix} = \frac{2}{m} \mathbf{X}^T \cdot (\mathbf{X} \cdot \theta - \mathbf{y})$$

Gradient Descent step

$$\theta^{(\text{next step})} = \theta - \eta \nabla_{\text{MSE}}(\theta)$$

Handwritten notes:

- 12
- X
- 500
- $\theta = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$



Not a linear closed formula

6.2. Calculate the MSE for the following example in spark

3.1.1 Estimating the coefficients (closed form, or, analytical solution)

Using calculus, one can show that $\hat{\beta}_0$ and $\hat{\beta}_1$ can be estimated as follows:

Least Squares Coefficients Estimates:
(equation 3.4)

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

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x},$$

where

$$\bar{y} \equiv \frac{1}{n} \sum_{i=1}^n y_i$$

$$\bar{x} \equiv \frac{1}{n} \sum_{i=1}^n x_i$$

are the sample means.

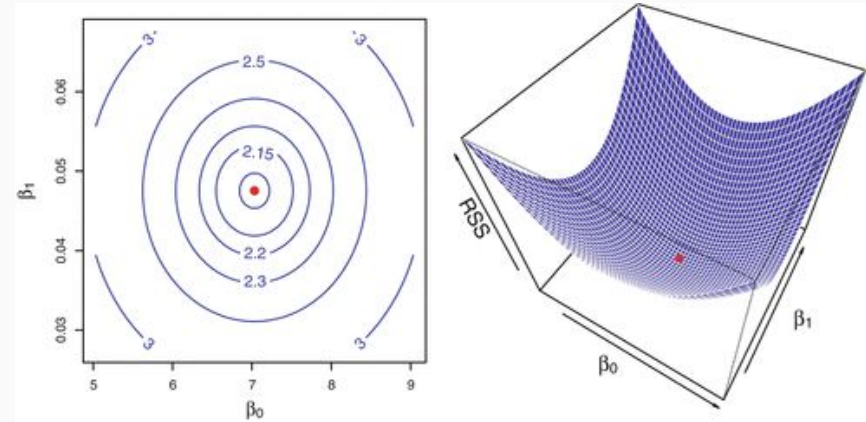


Fig. 3.2

Contour and three-dimensional plots of the RSS on the Advertising data, using sales as the response and TV as the predictor. The red dots correspond to the least squares estimates $\hat{\beta}_0$ and $\hat{\beta}_1$.

3.1.2 Assessing the Accuracy of the Coefficient Estimates

The *true* relationship between X and Y takes the form

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

for some unknown function f , where ϵ is a mean-zero random error term. If f is to be approximated by a linear function, then we can write this relationship as

$$Y = \beta_0 + \beta_1 X + \epsilon$$

Here β_0 is the intercept term—that is, the expected value of Y when $X = 0$, and β_1 is the slope. The model given by the above equation defines the *population regression line*, which is the best linear approximation to the true relationship between X and Y .

3.1.2 Assessing the Accuracy of the Coefficient Estimates

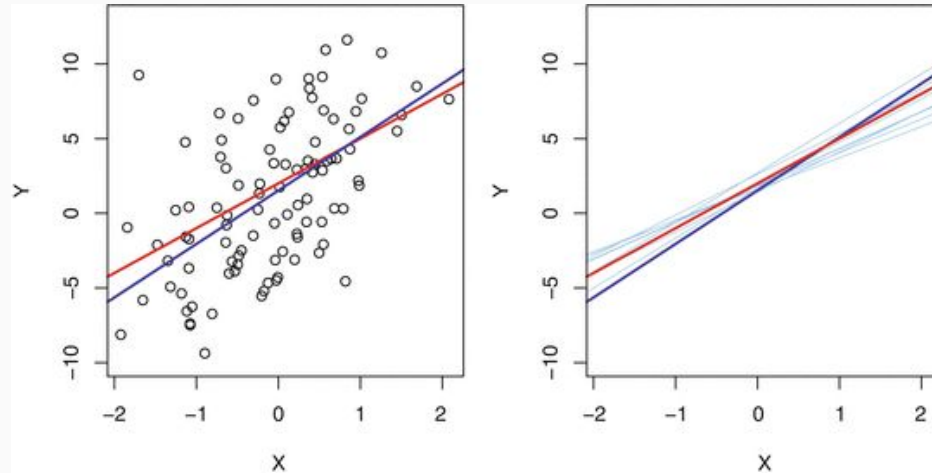


Fig. 3.3

A simulated data set. **Left:** The red line represents the true relationship, $f(X)=2+3X$, which is known as the population regression line. The blue line is the least squares line; it is the least squares estimate for $f(X)$ based on the observed data, shown in black. **Right:** The population regression line is again shown in red, and the least squares line in dark blue. In light blue, ten least squares lines are shown, each computed on the basis of a separate random set of observations. Each least squares line is different, but on average, the least squares lines are quite close to the population regression line.

Population regression line

$$Y = \beta_0 + \beta_1 X + \epsilon$$

Least squares line

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

3.1.2 Assessing the Accuracy of the Coefficient Estimates - unbiased estimator

The difference between the **population regression line** and the **least squares line** can be thought of in terms of the standard statistical approach of using information from a sample to estimate characteristics of a large population. For example, suppose we want to know the population mean μ for some random variable Y . μ is not known, but we can estimate μ from observations in a sample of the data:

$$\hat{\mu} = \bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$$

The analogy between Linear Regression and estimation of a mean is based on the concept of bias. The sample $\hat{\mu}$ is an unbiased estimate of μ . What this means is that on the basis of one particular set of observations y_1, \dots, y_n , $\hat{\mu}$ might overestimate μ , and on the basis of another set of observations, $\hat{\mu}$ might underestimate μ . But if we could average a huge number of estimates of μ obtained from a huge number of sets of observations, then this average would *exactly* equal μ . Hence, an **unbiased estimator** does not *systematically* over- or under- estimate the true parameter. The property of unbiasedness holds for the least squares coefficient estimates given by (3.4) as well: if we estimate β_0 and β_1 on the basis of a particular data set, then our estimates won't be exactly equal to β_0 and β_1 . But if we could average the estimates obtained over a huge number of data sets, then the average of these estimates would be spot on!

3.1.2 Assessing the Accuracy of the Coefficient Estimates - Standard Error

How accurate is the sample mean $\hat{\mu}$ as an estimate of μ ?

This question can be answered by calculating the *standard error* of $\hat{\mu}$, written as $SE(\hat{\mu})$. In addition, we can calculate how this deviation shrinks with n - the more observations we have, the smaller the standard error of $\hat{\mu}$:

$$\text{Var}(\hat{\mu}) = SE(\hat{\mu})^2 = \frac{\sigma^2}{n},$$

The same holds true for $\hat{\beta}_0$ and $\hat{\beta}_1$:

$$SE(\hat{\beta}_0)^2 = \sigma^2 \left[\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^n (x_i - \bar{x})^2} \right], \quad SE(\hat{\beta}_1)^2 = \frac{\sigma^2}{\sum_{i=1}^n (x_i - \bar{x})^2},$$

Where $\sigma^2 = \text{Var}(\varepsilon)$. In general, σ^2 is not known, but can be estimated from the data. This estimate of σ is known as the *residual standard error*:

$$RSE = \sqrt{RSS/(n - 2)}.$$

3.1.2 Assessing the Accuracy of the Coefficient Estimates - Confidence Intervals

Standard errors can be used to compute *confidence intervals*. A 95 % confidence interval is defined as a range of values such that with 95 % probability, the range will contain the true unknown value of the parameter. The range is defined in terms of lower and upper limits computed from the sample of data. For linear regression, the 95 % confidence interval for β_1 approximately takes the form

$$\hat{\beta}_1 \pm 2 \cdot \text{SE}(\hat{\beta}_1).$$

That is, there is approximately a 95 % chance that the interval:

$$[\hat{\beta}_1 - 2 \cdot \text{SE}(\hat{\beta}_1), \hat{\beta}_1 + 2 \cdot \text{SE}(\hat{\beta}_1)]$$

will contain the true value of β_1 . Similarly, a confidence interval for β_0 approximately takes the form:

$$\hat{\beta}_0 \pm 2 \cdot \text{SE}(\hat{\beta}_0).$$

3.1.2 Assessing the Accuracy of the Coefficient Estimates - t-statistic, p-value

Standard errors can also be used to perform *hypothesis tests* on the coefficients. The most common hypothesis test involves testing the *null hypothesis* of **H₀: There is no relationship between X and Y -> $\beta_1=0$** versus the *alternative hypothesis* **H_a: There is some relationship between X and Y -> $\beta_1 \neq 0$**

In practice, we compute a *t-statistic*, given by
$$t = \frac{\hat{\beta}_1 - 0}{SE(\hat{\beta}_1)},$$
 model error degrees of freedom = number of observations – number of parameters

which measures the number of standard deviations that $\hat{\beta}_1$ is away from 0. If there really is no relationship between X and Y, then we expect that t will have a *t-distribution* with $n - 2$ degrees of freedom. The *t-distribution* has a bell shape and for values of n greater than approximately 30 it is quite similar to the normal distribution. Consequently, it is a simple matter to compute the probability of observing any value equal to $|t|$ or larger, assuming $\beta_1 = 0$. We call this probability the *p-value*.

Once we have rejected the null hypothesis in favor of the alternative hypothesis, it is natural to want to quantify *the extent to which the model fits the data*. The quality of a linear regression fit is typically assessed using two related quantities: the *residual standard error* (RSE) and the R^2 statistic.

3.1.3 Assessing the Accuracy of the Model - Residual Standard Error

The Residual Standard Error (RSE) is an estimate of the standard deviation of ε . Roughly speaking, it is the average amount that the response will deviate from the true regression line. It is computed using the formula:

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

The RSE is considered a measure of the **lack of fit** of the model to the data. If the predictions obtained using the model are very close to the true outcome values—that is, if $\hat{y}_i \approx y_i$ for $i = 1, \dots, n$ —then RSE will be small, and we can conclude that the model fits the data very well. On the other hand, if \hat{y}_i is very far from y_i for one or more observations, then the RSE may be quite large, indicating that the model doesn't fit the data well.

3.1.3 Assessing the Accuracy of the Model - R^2 statistic

The RSE provides an absolute measure of lack of fit of the model to the data. But since it is measured in the units of Y , it is not always clear what constitutes a good RSE. The R^2 statistic provides an alternative measure of fit. It takes the form of a *proportion*—the proportion of variance explained—and so it always takes on a value between 0 and 1, and is independent of the scale of Y . To calculate R^2 we use the formula

$$R^2 = \frac{\text{TSS} - \text{RSS}}{\text{TSS}} = 1 - \frac{\text{RSS}}{\text{TSS}}$$

where $\text{TSS} = \sum (y_i - \bar{y})^2$ is the *total sum of squares*. TSS measures the total variance in the response Y , and can be thought of as the amount of variability inherent in the response before the regression is performed.

The R^2 statistic has an interpretational advantage over the RSE, since unlike the RSE, it always lies between 0 and 1. However, it can still be challenging to determine what is a *good* R^2 value, and in general, this will depend on the application.

3.1.3 Assessing the Accuracy of the Model - correlation

The R^2 statistic is a measure of the linear relationship between X and Y . Recall that *correlation*, defined as

$$\text{Cor}(X, Y) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}}$$

is also a measure of the linear relationship between X and Y . This suggests that we might be able to use $r = \text{Cor}(X, Y)$ instead of R^2 in order to assess the fit of the linear model. In fact, it can be shown that in the simple linear regression setting, $R^2 = r^2$.

The concept of correlation between the predictors and the response does not extend automatically to multiple linear regression, since correlation quantifies the association between a single pair of variables rather than between a larger number of variables. We will see that R^2 fills this role in multiple linear regression.

3.2 Multiple Linear Regression

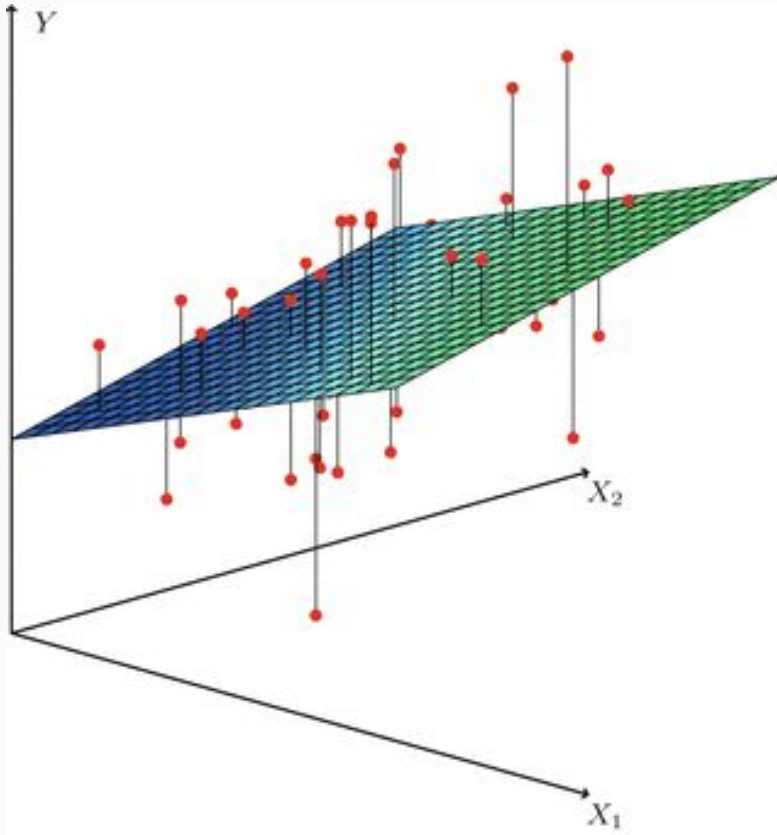


Fig. 3.4

In a three-dimensional setting, with two predictors and one response, the least squares regression line becomes a plane. The plane is chosen to minimize the sum of the squared vertical distances between each observation (shown in red) and the plane.

3.2.2 Important questions for multiple linear regression

1. *Is at least one of the predictors X_1, X_2, \dots, X_p useful in predicting the response?*
2. *Do all the predictors help to explain Y , or is only a subset of the predictors useful?*
3. *How well does the model fit the data?*
4. *Given a set of predictor values, what response value should we predict, and how accurate is our prediction?*

Q1: Is There a Relationship Between the Response and Predictors?

This hypothesis test is performed by computing the *F*-statistic

$$F = \frac{(\text{TSS} - \text{RSS})/p}{\text{RSS}/(n - p - 1)};$$

Q2: Deciding on Important Variables

The task of determining which predictors are associated with the response, in order to fit a single model involving only those predictors, is referred to as *variable selection*.

- Mallow's C_p
- Akaike information criterion
- Bayesian information criterion
- adjusted R^2
- forward selection
- null model
- backward selection
- mixed selection

Q3: Model fit

Two of the most common numerical measures of model fit are the RSE and R^2 , the fraction of variance explained. These quantities are computed and interpreted in the same fashion as for simple linear regression.

$$\text{RSE} = \sqrt{\frac{1}{n - p - 1} \text{RSS}},$$

$$\text{RSS} = e_1^2 + e_2^2 + \cdots + e_n^2$$

$$e_i = y_i - \hat{y}_i$$

$$R^2 = \frac{\text{TSS} - \text{RSS}}{\text{TSS}} = 1 - \frac{\text{RSS}}{\text{TSS}}$$

$$\text{TSS} = \sum (y_i - \bar{y})^2$$

Q4: Predictions

- *reducible error/model bias*
- *irreducible error*

Prediction intervals are always wider than confidence intervals, because they incorporate both the error in the estimate for $f(X)$ (*the reducible error*) and the uncertainty as to how much an individual point will differ from the population regression plane (*the irreducible error*).

3.3 Other Considerations in the Regression Model

✓ 3.3 Other Considerations in the Regression Model

✓ 3.3.1 Qualitative Predictors

 Predictors with Only Two Levels

 Qualitative Predictors with More than Two Levels

✓ 3.3.2 Extensions of the Linear Model

 Removing the Additive Assumption

 Non-linear Relationships

✓ 3.3.3 Potential Problems

 1. Non-linearity of the Data

 2. Correlation of Error Terms

 3. Non-constant Variance of Error Terms

 4. Outliers

 5. High Leverage Points

 6. Collinearity