
Data Science
Regression
Analysis
- DIY for Newbies

Data Science - Regression Analysis (DIY for newbies)

Includes 100 solved problems

Geetha Subramanian

This book is for sale at http://leanpub.com/Regressionanalysis

This version was published on 2020-07-18

DEDICATED TO MY DAUGHTER HARSHITHA

Contents

Supervised Learning Algorithms:

Supervised algorithm is used when we have input(x) and the output(y) variable and we need to develop a model where output is a function of input.

$y = f(x)$

Supervised learning algorithm can be used for regression and classification problems.

In this book, we will focus on regression models.

Let us explore them

Regression

Regression is used when the output(y) variable is a real number. It is used to predict sale price of houses, height or weight of people, salary of employees etc.

There are different regression methods used to predict y.

*Simple Linear Regression

*Polynomial Regression

*Support Vector Regression

*Decision Tree Regression

*Random Forest Regression

These are various regression models. Now, we will look into creating basic models in R and python using these models. Idea behind this book is to introduce the basic coding techniques for different models. Tuning of different parameters in the codes will be discussed in the next level.

Simple Linear Regression

In Simple Linear Regression output variable(Y) depends on one input variable (X). The formula for simple linear regression is

y = a+bx

Where a is the intercept and b is the slope. The aim is to predict the value of y with this equation. The line which fits this equation is known as the line of best fit , where the sum of squared residuals(difference between actual and predicted values of Y) is minimum.

Linear regression problems can be solved through Python and R. Two standard formats are attached for solving any linear regression problems through Python and R

Python Programming language - Linear Regression

Predict Marks scored based on hours studied. Here Marks is y (dependent variable/target) and Hours studied is X (independent variable).

```
#load numpy and panda libraries
import pandas as pd
import numpy as np


#Create a dataset
data = pd.DataFrame({'Hours studied':[0,1,2,3,4,5,6,7,8,9,10],'Marks':[10,40,50,60,7\
0,75,80,85,90,95,100]})


#segregate X and y details
X= data.iloc[:, 0:1].values
y= data.iloc[:, 1].values


#split the dataset into train and test data
from sklearn.model_selection import train_test_split
X_train, X_test, y_train,y_test = train_test_split(X,y,test_size = 1/3, random_state\
 =0)


#load linear regression
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()


#fit linear regression model on train data
regressor.fit(X_train,y_train)


#predict the values for the test results
y_pred = regressor.predict(X_test)


#check the accuracy of result using root meanuaqaured error
from sklearn.metrics import mean_squared_error
import math

```

```
39
40 slr_rmse = math.sqrt(mean_squared_error(y_test,y_pred))
41
42 slr_rmse
```

R Programming Language - Linear Regression

Predict Marks scored based on hours studied. Here Marks is y (dependent variable/target) and Hours studied is X (independent variable).

```r
library(caTools)


#create a dataset
df <- data.frame('HoursStudied' = c(0,1,2,3,4,5,6,7,8,9,10),
                 'Marks' = c(10,40,50,60,70,75,80,85,90,95,100))


#Create training data and test data
split <- sample.split(df$Marks, SplitRatio = 2/3)

training_set <- subset(df, split == TRUE)

test_set <- subset(df, split == FALSE)

#Fitting linear regression to the training data
regressor  <- lm(Marks~HoursStudied,
                 data = training_set)

summary(regressor)


#predicting the results
y_pred <- predict(regressor, newdata = test_set)

#Find the accuracy
rmse <-sqrt(mean((test_set$Marks - y_pred)^2))

print(rmse)
```

Polynomial Regression

Polynomial regression is used when there is a correlation between the variables, however the relation doesn't look linear.

Example of a polynomial regression equation:

$y = a + a1X1 + a2(X1)^2$

To find a linear relation we have taken square values of X1. Once the values are transformed, they would make a linear relation.

Python Programming language - Polynomial Regression

Predict salary based on Grade.

```
#create dataset with Grade and corresponding salary
import pandas as pd

data = pd.DataFrame({'Grade':[1,2,3,4,5,6,7,8,9,10],
                     'Salary':[100,100,200,300,400,450,500,500,550,600]})

#define X and y
X = data[['Grade']].values
y = data['Salary'].values


#split the data into train and test
from sklearn.model_selection import train_test_split


X_train, X_test, y_train,y_test = train_test_split(X,y,test_size = 1/3, random_state\
 =0)


#set the polynomial degree to 3 and convert the x values
from sklearn.preprocessing import PolynomialFeatures


p_r = PolynomialFeatures(degree=3)


X_train_poly = p_r.fit_transform(X_train)


X_test_poly = p_r.fit_transform(X_test)

#define the model and fit the model
from sklearn.linear_model import LinearRegression


model = LinearRegression()


model = model.fit(X_train_poly,y_train)
```

```
40
41 #predict salary for Grade 7.5
42 y_pred = model.predict(X_test_poly)
43
44
45 poly_rmse = math.sqrt(mean_squared_error(y_test,y_pred))
46
47
48 poly_rmse
```

R Programming Language - Polynomial regression

Predict salary based on Grade.

```r
#Load the data
data <- data.frame('Grade'= c(1,2,3,4,5,6,7,8,9,10),
                   'Salary'= c(100,100,200,300,400,450,500,500,550,600))

#add polynomial features
data$Grade_2 <- data$Grade^2

data$Grade_3 <- data$Grade^3

#Create training data and test data
split <- sample.split(data$Salary, SplitRatio = 2/3)

training_set <- subset(data, split == TRUE)

test_set <- subset(data, split == FALSE)

#develop the polynomial regression model
poly_reg <-lm(Salary~., data = training_set)

#predict the value for y, when Grade is 6.5
y_pred <- predict(poly_reg, newdata = test_set)

y_pred

#Find the accuracy
rmse <-sqrt(mean((test_set$Salary - y_pred)^2))

print(rmse)
```

Support Vector Regression

Support Vector regression supports linear and non-linear regression.

Python Programming language - Support Vector Regression

Predict salary based on Grade.

```
#load the dataset
data = pd.DataFrame({'Grade':[1,2,3,4,5,6,7,8,9,10],
                     'Salary':[100,100,200,300,400,450,500,500,550,600]})

#identify X and y values
X = data[['Grade']].values

y = data['Salary'].values


#split the data into train and test
from sklearn.model_selection import train_test_split

X_train, X_test, y_train,y_test = train_test_split(X,y,test_size = 1/3, random_state\
 =12)


#perfrorm feature scaling on the data
from sklearn.preprocessing import StandardScaler

sc_X = StandardScaler()

sc_y = StandardScaler()

X_train = sc_X.fit_transform(X_train)

y_train = np.squeeze(sc_y.fit_transform(y_train.reshape(-1, 1)))

X_test = sc_X.fit_transform(X_test)

y_test = np.squeeze(sc_y.fit_transform(y_test.reshape(-1, 1)))


#fit model on the datat
from sklearn.svm import SVR

regressor = SVR(kernel = 'rbf')

regressor.fit(X_train, y_train)
```

```
40
41
42  #predict the results
43  y_pred = regressor.predict(X_test)
44
45  y_pred = sc_y.inverse_transform(y_pred)
46
47
48  #compute the accuracy
49  y_test = sc_y.inverse_transform(y_test)
50
51  svm_mse = mean_squared_error(y_test,y_pred)
52
53
54  svm_mse
```

R Programming Language - Support Vector regression

Predict salary based on Grade.

```
#load the libraru
library(caret)


#Load the data
data <- data.frame('Grade'= c(1,2,3,4,5,6,7,8,9,10),
                   'Salary'= c(100,100,200,300,400,450,500,500,550,600))


#Create training data and test data
data_p <- createDataPartition(y = data$Salary, p = 0.8, list = FALSE, groups = 2)

train_data <- data[data_p,]

test_data <- data[-data_p,]

#develop svm model
trctrl = trainControl(method = 'repeatedcv', number = 5, repeats = 2)

model <- train(Salary~., data = train_data, method = 'svmRadial', trainControl = trc\
trl)

model

#predict the result
y_pred <- predict(model,test_data)

y_pred

#check the accuracy

mse <-mean((test_data$Salary - y_pred)^2)

print(mse)
```

Decision Tree Regression:

Decision Tree is a tree based model with a root node which is the starting point of the tree and leaf node/terminal node is the final point where the decision is made on the class or value. There are many Decision nodes in the middle which helps in splitting the data for arriving at a prediction.

Decision Tree uses Entropy and Gini to partition the data into subsets to include instances with similar values

Python Programming language - Decision Tree Regression

Predict salary based on Grade.

```
#load the dataset
import pandas as pd

data = pd.DataFrame({'Grade':[1,2,3,4,5,6,7,8,9,10],
                     'Salary':[100,100,200,300,400,450,500,500,550,600]})

#identify X and y values
X = data[['Grade']].values

y = data['Salary'].values

#split the data into train and test
from sklearn.model_selection import train_test_split

X_train, X_test, y_train,y_test = train_test_split(X,y,test_size = 1/3, random_state\
 =10)

#build the regressor model
from sklearn.tree import DecisionTreeRegressor

regressor = DecisionTreeRegressor(random_state = 0)
regressor.fit(X_train,y_train)

#predict the results
y_pred = regressor.predict(X_test)

y_pred

#compute the accuracy
from sklearn.metrics import mean_absolute_error

mae = mean_absolute_error(y_test,y_pred)

mae
```

R Programming Language - Decision Tree Regression

Predict salary based on Grade.

```r
#load the library
library(caret)


#Load the data
data <- data.frame('Grade'= c(1,2,3,4,5,6,7,8,9,10),
                   'Salary'= c(100,100,200,300,400,450,500,500,550,600))


#Create training data and test data
data_p <- createDataPartition(y = data$Salary, p = 0.7, list = FALSE, groups = 2)

train_data <- data[data_p,]

test_data <- data[-data_p,]

#develop the decision tree regressor model
library(rpart)

regressor <- rpart(formula = Salary ~., data = train_data)

#predict the result
y_pred <- predict(regressor, test_data)

y_pred

#check the accuracy
mae <-mean((test_data$Salary - y_pred))

print(mae)
```

Random Forest Regression:

A Random Forest is an ensemble technique capable of performing regression tasks with the use of multiple decision trees and a technique called Bootstrap Aggregation, commonly known as bagging. Bagging involves training each decision tree on a different data sample. Many decision trees are built in Random Forest, which makes the prediction more accurate.

Python Programming language - Random Forest Regression

Predict salary based on Grade.

```
#load the libraries
import numpy as np
import pandas as pd
import sklearn

#load the file
dataset = pd.read_csv('file.csv')

#identify X and Y columns
X= dataset.iloc[:, 1:2].values

y= dataset.iloc[:, 2].values


#build the randon forest regressor model
from sklearn.ensemble import RandomForestRegressor

regressor = RandomForestRegressor(n_estimators = 100, random_state = 0)

regressor.fit(X,y)

#predict the results
y_pred = regressor.predict([[6.5]])
```

R Programming Language - Random Forest Regression

Predict salary based in Grade.

```
#load the file
dataset <- read.csv("file.csv")

#build the Random Forest Regression
library(randomForest)
set.seed(1234)
regressor <- randomForest(x = dataset[1],y = dataset$Salary,
                          ntree = 500)


#Predict the result
y_pred <- predict(regressor, data.frame(Level = 6.5))
```

100 Solved Questions

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q1 (Python - Prediction with Linear Regression)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q2 (Python - Linear Regression, PCA, MSE)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q3 (R - Prediction with Linear model)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q4 (Python - Jaccod Index)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q5 (R - Stochastic Gradient Boosting)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q6 (Python - Boxplot)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q7 (Python - Sort using attrgetter)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q8 (Python - Heatmap)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q9 (Python - Euclidean Distance)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q10 (Python - Manhattan Distance)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q11 (R - Bar Plot)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q12 (R - Formatting Date)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q13 (Python - Forward selection)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q14 - (Linear Model)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q15 (Python - Correlation)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q16 (Python - Manhattan Distance)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q17 (Python - Word Split)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q18 (Python - PCA)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q19 (Python - Eigenvector)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q20 (Python - Eigenvalues)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q21 (R - Linear Regression)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q22 (Python - OLS)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q23 (Python - OLS)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q24 (Python - Decision Tree)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q25 (R - Random Forest)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q26 (Python - Text)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q27 (Python - Sort)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q28 (Python - Indexing)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q29 (Python - OLS)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q30 (Python - OLS)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q31 (Python - Chi-Square)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q32 (Python - PCA)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q33 (Python - Variance Inflation Factor)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q34 (Python - Slope and P-Value)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q35 (Python - Recusive Feature)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q36 (R - Random Forest)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q37 (R - Support Vector Regression)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q38 (R - Polynomial Regression)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q39 (R - Linear Regression)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q40 (Python - Chi value)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q41 (R - Attribute Importance)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q42 (R - Baggging model)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q43 (R - Stacking Algorithm)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q44 (R - PCA)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q45 (Python)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q46 (Python - Random Forest)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q47 (R - Forward selection)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q48 (R - Support Vector Regressor)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q49 (R - Polynomial Regression)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q50 (R - Mean, Median, Mode)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q51 (Python - Correlation coefficients)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q52 (R - Linear Regression)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q53 (R - Linear Regression)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q54 (Python - Text)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q55 (R - Polynomial Regression)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q56 (R - Support Vector Regression)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q57 (R - Decision Tree)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q58 (R - Random Forest)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q59 (Python - Outliers)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q60 (Python - PCA)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q61 (Python - Ridge Regression)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q62 (Python - Lasso)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q63 (Python - Impute)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q64 (Python - Impute)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q65 (Python - Support Vector Regression)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q66 (Python - Decision Tree)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q67 (Python - Random Forest)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q68 (Python- Mean, Median, Mode,IQR)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q69 (R - Outliers)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q70 (Python - Outlier)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q71 (Python - Decision Tree)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q72 (R - Decision Tree)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q73 (R - Ridge Regression)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q74 (R - Ridge Regression)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q75 (R - Ridge Regression)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q76 (Python - Ridge Regression)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q77 (Python - Ridge Regression)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q78 (Python - Ridge Regression)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q79 (R - Lasso)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q80 (R - Lasso)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q81 (R - Lasso)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q82 (Python - Lasso)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q83 (Python - Lasso)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q84 (Python - Lasso)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q85 (R - IQR)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q86 (Python - IQR)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q87 (R - Linear)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q88 (Python - Linear Regression)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q89 (R - Random Forest)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q90 (R - Random Forest)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q91 (Python - Random Forest)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q92 (Python - Random Forest)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q93 (Python - Polynomial Regression)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q94 (Python - Impute)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q95 (R - Impute)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q96 (R - Impute)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q97 (Python - Linear)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q98 (Python - Adaboost)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q99 (Calculation - Slope and Intercept)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Q100 (Python - Backward Elimination)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.

Appendix:

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/Regressionanalysis.