



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



This is a [Leanpub](#) book. Leanpub empowers authors and publishers with the Lean Publishing process. [Lean Publishing](#) is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

© 2020 Geetha Subramanian

DEDICATED TO MY DAUGHTER HARSHITHA

Contents

Supervised Learning Algorithms:	1
Regression	2
Polynomial Regression	7
Support Vector Regression	11
Decision Tree Regression:	15
Random Forest Regression:	18
100 Solved Questions	21
Q1 (Python - Prediction with Linear Regression)	21
Q2 (Python - Linear Regression, PCA, MSE)	21
Q3 (R - Prediction with Linear model)	21
Q4 (Python - Jaccod Index)	21
Q5 (R - Stochastic Gradient Boosting)	21
Q6 (Python - Boxplot)	21
Q7 (Python - Sort using attrgetter)	22
Q8 (Python - Heatmap)	22
Q9 (Python - Euclidean Distance)	22
Q10 (Python - Manhattan Distance)	22
Q11 (R - Bar Plot)	22
Q12 (R - Formatting Date)	22
Q13 (Python - Forward selection)	22
Q14 - (Linear Model)	23
Q15 (Python - Correlation)	23
Q16 (Python - Manhattan Distance)	23
Q17 (Python - Word Split)	23
Q18 (Python - PCA)	23
Q19 (Python - Eigenvector)	23
Q20 (Python - Eigenvalues)	23
Q21 (R - Linear Regression)	24
Q22 (Python - OLS)	24
Q23 (Python - OLS)	24
Q24 (Python - Decision Tree)	24
Q25 (R - Random Forest)	24
Q26 (Python - Text)	24

CONTENTS

Q27 (Python - Sort)	24
Q28 (Python - Indexing)	25
Q29 (Python - OLS)	25
Q30 (Python - OLS)	25
Q31 (Python - Chi-Square)	25
Q32 (Python - PCA)	25
Q33 (Python - Variance Inflation Factor)	25
Q34 (Python - Slope and P-Value)	25
Q35 (Python - Recursive Feature)	26
Q36 (R - Random Forest)	26
Q37 (R - Support Vector Regression)	26
Q38 (R - Polynomial Regression)	26
Q39 (R - Linear Regression)	26
Q40 (Python - Chi value)	26
Q41 (R - Attribute Importance)	26
Q42 (R - Bagging model)	27
Q43 (R - Stacking Algorithm)	27
Q44 (R - PCA)	27
Q45 (Python)	27
Q46 (Python - Random Forest)	27
Q47 (R - Forward selection)	27
Q48 (R - Support Vector Regressor)	27
Q49 (R - Polynomial Regression)	28
Q50 (R - Mean, Median, Mode)	28
Q51 (Python - Correlation coefficients)	28
Q52 (R - Linear Regression)	28
Q53 (R - Linear Regression)	28
Q54 (Python - Text)	28
Q55 (R - Polynomial Regression)	28
Q56 (R - Support Vector Regression)	29
Q57 (R - Decision Tree)	29
Q58 (R - Random Forest)	29
Q59 (Python - Outliers)	29
Q60 (Python - PCA)	29
Q61 (Python - Ridge Regression)	29
Q62 (Python - Lasso)	29
Q63 (Python - Impute)	30
Q64 (Python - Impute)	30
Q65 (Python - Support Vector Regression)	30
Q66 (Python - Decision Tree)	30
Q67 (Python - Random Forest)	30
Q68 (Python- Mean, Median, Mode,IQR)	30
Q69 (R - Outliers)	30

CONTENTS

Q70 (Python - Outlier)	31
Q71 (Python - Decision Tree)	31
Q72 (R - Decision Tree)	31
Q73 (R - Ridge Regression)	31
Q74 (R - Ridge Regression)	31
Q75 (R - Ridge Regression)	31
Q76 (Python - Ridge Regression)	31
Q77 (Python - Ridge Regression)	32
Q78 (Python - Ridge Regression)	32
Q79 (R - Lasso)	32
Q80 (R - Lasso)	32
Q81 (R - Lasso)	32
Q82 (Python - Lasso)	32
Q83 (Python - Lasso)	32
Q84 (Python - Lasso)	33
Q85 (R - IQR)	33
Q86 (Python - IQR)	33
Q87 (R - Linear)	33
Q88 (Python - Linear Regression)	33
Q89 (R - Random Forest)	33
Q90 (R - Random Forest)	33
Q91 (Python - Random Forest)	34
Q92 (Python - Random Forest)	34
Q93 (Python - Polynomial Regression)	34
Q94 (Python - Impute)	34
Q95 (R - Impute)	34
Q96 (R - Impute)	34
Q97 (Python - Linear)	34
Q98 (Python - Adaboost)	35
Q99 (Calculation - Slope and Intercept)	35
Q100 (Python - Backward Elimination)	35
Appendix:	36

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).

```
1  #load numpy and panda libraries
2  import pandas as pd
3  import numpy as np
4
5
6  #Create a dataset
7  data = pd.DataFrame({'Hours studied':[0,1,2,3,4,5,6,7,8,9,10], 'Marks':[10,40,50,60,7\
8  0,75,80,85,90,95,100]})
9
10
11 #segregate X and y details
12 X= data.iloc[:, 0:1].values
13 y= data.iloc[:, 1].values
14
15
16 #split the dataset into train and test data
17 from sklearn.model_selection import train_test_split
18 X_train, X_test, y_train,y_test = train_test_split(X,y,test_size = 1/3, random_state\
19     =0)
20
21
22 #load linear regression
23 from sklearn.linear_model import LinearRegression
24 regressor = LinearRegression()
25
26
27 #fit linear regression model on train data
28 regressor.fit(X_train,y_train)
29
30
31 #predict the values for the test results
32 y_pred = regressor.predict(X_test)
33
34
35 #check the accuracy of result using root meanuaqaured error
36 from sklearn.metrics import mean_squared_error
37 import math
38
```

```
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).

```
1  library(caTools)
2
3
4  #create a dataset
5  df <- data.frame('HoursStudied' = c(0,1,2,3,4,5,6,7,8,9,10),
6                    'Marks' = c(10,40,50,60,70,75,80,85,90,95,100))
7
8
9  #Create training data and test data
10 split <- sample.split(df$Marks, SplitRatio = 2/3)
11
12 training_set <- subset(df, split == TRUE)
13
14 test_set <- subset(df, split == FALSE)
15
16 #Fitting linear regression to the training data
17 regressor <- lm(Marks~HoursStudied,
18                 data = training_set)
19
20 summary(regressor)
21
22
23 #predicting the results
24 y_pred <- predict(regressor, newdata = test_set)
25
26 #Find the accuracy
27 rmse <- sqrt(mean((test_set$Marks - y_pred)^2))
28
29 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:

$$y = a + a_1X_1 + a_2(X_1)^2$$

To find a linear relation we have taken square values of X_1 . Once the values are transformed, they would make a linear relation.

Python Programming language - Polynomial Regression

Predict salary based on Grade.

```
1  #create dataset with Grade and corresponding salary
2  import pandas as pd
3
4  data = pd.DataFrame({'Grade': [1,2,3,4,5,6,7,8,9,10],
5                       'Salary': [100,100,200,300,400,450,500,500,550,600]})
6
7  #define X and y
8  X = data[['Grade']].values
9  y = data['Salary'].values
10
11
12 #split the data into train and test
13 from sklearn.model_selection import train_test_split
14
15
16 X_train, X_test, y_train,y_test = train_test_split(X,y,test_size = 1/3, random_state\
17     =0)
18
19
20 #set the polynomial degree to 3 and convert the x values
21 from sklearn.preprocessing import PolynomialFeatures
22
23
24 p_r = PolynomialFeatures(degree=3)
25
26
27 X_train_poly = p_r.fit_transform(X_train)
28
29
30 X_test_poly = p_r.fit_transform(X_test)
31
32 #define the model and fit the model
33 from sklearn.linear_model import LinearRegression
34
35
36 model = LinearRegression()
37
38
39 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.

```
1  #Load the data
2  data <- data.frame('Grade' = c(1,2,3,4,5,6,7,8,9,10),
3                      'Salary' = c(100,100,200,300,400,450,500,500,550,600))
4
5  #add polynomial features
6  data$Grade_2 <- data$Grade^2
7
8  data$Grade_3 <- data$Grade^3
9
10 #Create training data and test data
11 split <- sample.split(data$Salary, SplitRatio = 2/3)
12
13 training_set <- subset(data, split == TRUE)
14
15 test_set <- subset(data, split == FALSE)
16
17 #develop the polynomial regression model
18 poly_reg <- lm(Salary~., data = training_set)
19
20 #predict the value for y, when Grade is 6.5
21 y_pred <- predict(poly_reg, newdata = test_set)
22
23 y_pred
24
25 #Find the accuracy
26 rmse <- sqrt(mean((test_set$Salary - y_pred)^2))
27
28 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.

```
1  #load the dataset
2  data = pd.DataFrame({'Grade': [1,2,3,4,5,6,7,8,9,10],
3                        'Salary': [100,100,200,300,400,450,500,500,550,600]})
4
5  #identify X and y values
6  X = data[['Grade']].values
7
8  y = data['Salary'].values
9
10
11 #split the data into train and test
12 from sklearn.model_selection import train_test_split
13
14 X_train, X_test, y_train,y_test = train_test_split(X,y,test_size = 1/3, random_state\
15     =12)
16
17
18 #perform feature scaling on the data
19 from sklearn.preprocessing import StandardScaler
20
21 sc_X = StandardScaler()
22
23 sc_y = StandardScaler()
24
25 X_train = sc_X.fit_transform(X_train)
26
27 y_train = np.squeeze(sc_y.fit_transform(y_train.reshape(-1, 1)))
28
29 X_test = sc_X.fit_transform(X_test)
30
31 y_test = np.squeeze(sc_y.fit_transform(y_test.reshape(-1, 1)))
32
33
34 #fit model on the datat
35 from sklearn.svm import SVR
36
37 regressor = SVR(kernel = 'rbf')
38
39 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.

```
1  #load the libraru
2  library(caret)
3
4
5  #Load the data
6  data <- data.frame('Grade' = c(1,2,3,4,5,6,7,8,9,10),
7                      'Salary' = c(100,100,200,300,400,450,500,500,550,600))
8
9
10 #Create training data and test data
11 data_p <- createDataPartition(y = data$Salary, p = 0.8, list = FALSE, groups = 2)
12
13 train_data <- data[data_p,]
14
15 test_data <- data[-data_p,]
16
17 #develop svm model
18 trctrl = trainControl(method = 'repeatedcv', number = 5, repeats = 2)
19
20 model <- train(Salary~., data = train_data, method = 'svmRadial', trainControl = trc\
21 trl)
22
23 model
24
25 #predict the result
26 y_pred <- predict(model,test_data)
27
28 y_pred
29
30 #check the accuracy
31
32 mse <- mean((test_data$Salary - y_pred)^2)
33
34 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.

```
1  #load the dataset
2  import pandas as pd
3
4  data = pd.DataFrame({'Grade': [1,2,3,4,5,6,7,8,9,10],
5                        'Salary': [100,100,200,300,400,450,500,500,550,600]})
6
7  #identify X and y values
8  X = data[['Grade']].values
9
10 y = data['Salary'].values
11
12 #split the data into train and test
13 from sklearn.model_selection import train_test_split
14
15 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 1/3, random_state\
16     =10)
17
18 #build the regressor model
19 from sklearn.tree import DecisionTreeRegressor
20
21 regressor = DecisionTreeRegressor(random_state = 0)
22 regressor.fit(X_train, y_train)
23
24 #predict the results
25 y_pred = regressor.predict(X_test)
26
27 y_pred
28
29 #compute the accuracy
30 from sklearn.metrics import mean_absolute_error
31
32 mae = mean_absolute_error(y_test, y_pred)
33
34 mae
```

R Programming Language - Decision Tree Regression

Predict salary based on Grade.

```
1  #load the library
2  library(caret)
3
4
5  #Load the data
6  data <- data.frame('Grade' = c(1,2,3,4,5,6,7,8,9,10),
7                      'Salary' = c(100,100,200,300,400,450,500,500,550,600))
8
9
10 #Create training data and test data
11 data_p <- createDataPartition(y = data$Salary, p = 0.7, list = FALSE, groups = 2)
12
13 train_data <- data[data_p,]
14
15 test_data <- data[-data_p,]
16
17 #develop the decision tree regressor model
18 library(rpart)
19
20 regressor <- rpart(formula = Salary ~., data = train_data)
21
22 #predict the result
23 y_pred <- predict(regressor, test_data)
24
25 y_pred
26
27 #check the accuracy
28 mae <- mean((test_data$Salary - y_pred))
29
30 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.

```
1  #load the libraries
2  import numpy as np
3  import pandas as pd
4  import sklearn
5
6  #load the file
7  dataset = pd.read_csv('file.csv')
8
9  #identify X and Y columns
10 X= dataset.iloc[:, 1:2].values
11
12 y= dataset.iloc[:, 2].values
13
14
15 #build the random forest regressor model
16 from sklearn.ensemble import RandomForestRegressor
17
18 regressor = RandomForestRegressor(n_estimators = 100, random_state = 0)
19
20 regressor.fit(X,y)
21
22 #predict the results
23 y_pred = regressor.predict([[6.5]])
```

R Programming Language - Random Forest Regression

Predict salary based in Grade.

```
1  #load the file
2  dataset <- read.csv("file.csv")
3
4  #build the Random Forest Regression
5  library(randomForest)
6  set.seed(1234)
7  regressor <- randomForest(x = dataset[,1], y = dataset$Salary,
8                             ntree = 500)
9
10
11 #Predict the result
12 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 - Recursive 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 - Bagging 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>.