Rapport du MOOC

Data Analysis and Visualization

Author: Mohamed Sneiba HAMOUD

Supervisor: Thomas LIM

Table of Contents

Why I chose this MOOC

Data and visual analytics is an emerging field concerned with analyzing, modeling, and visualizing complex high dimensional data. It allows us to extract knowledge and insights from data in various forms.
This is a must have skill for whoever aspires to become a data scientist like me. This course will allow me to enlarge my skills in this particular area, that I am really interested in.

Organization of this course

This course is mainly divided into three parts:

1. R programming language: This part is dedicated to help students master the R programming language with exercices on libraries, different data strutures, controls, function, ….
2. Data analysis: In this part, we focused on the format of the data (missing values, outliers, …) in order to preprocess the data before modeling it. We also used different visualization technique in order to get a first insight on the data.
3. Regression: This is the part where we start building models in order to predict future outcomes. We mainly used logistic regression and linear regression. We also studied different regularization methods to avoid overfitting.

Data Visualization in R

Data visualization is the creation and the study of the visual representation of data. In order to get some information, we use statistical graphics, plots and other techniques. Effective visualization helps us analyze and reason about data and evidence. It makes complex data more accessible, understandable and usable.
In this section, we will see the different techniques available in R to make visualization.

- Strip plots: It is a form of scatter plots using only 1 expression. When a data set is too large, plotting every instance of a dimension over a single metric can be a useful exercise for reviewing a distribution and discovering outliers in the data.

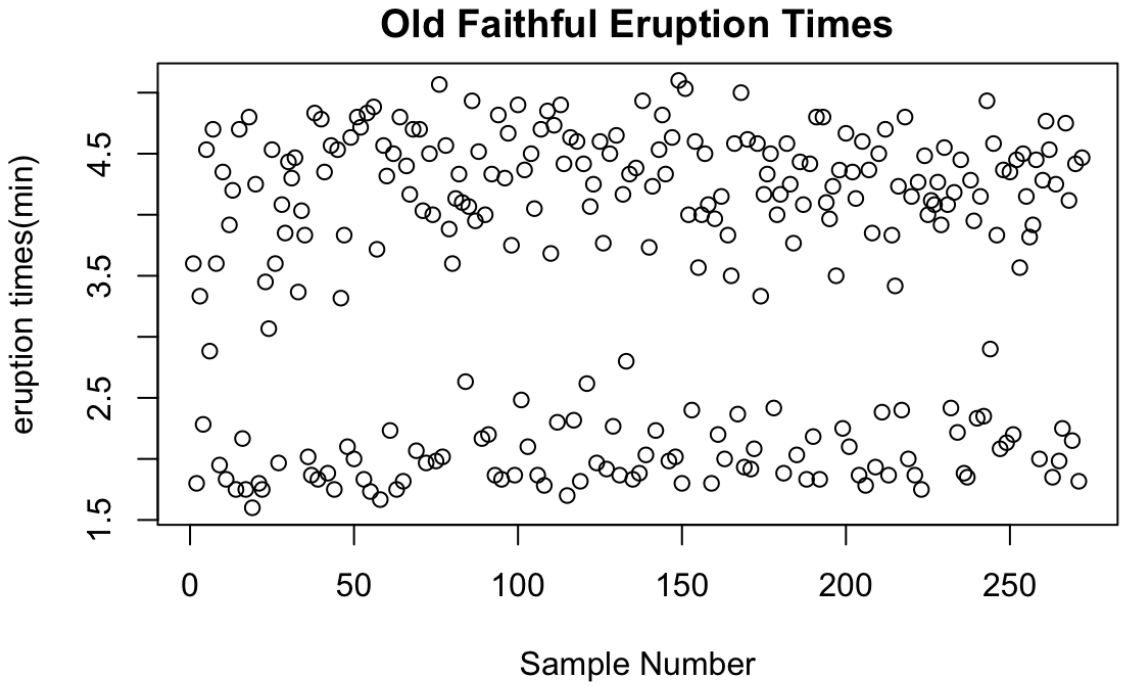


Figure 1 - Strip plot of the eruption times of of the old Faithful geyser, Wyoming, USA

In this strip plot, we can see that the old Faithful has two typical eruption times.

- Histograms: It graphs one-dimensional numeric data by dividing the range into bins width and counting the number of occurences in each bin.
  The width of the bins influences the level of detail of the plot. Very narrow bins maintain all the information present in the data but are hard to draw conclusions from, as the histogram become more equivalent to a sorted list of data values. Very wide bins lose informations to overly aggressive smoothing. A good bin width balances information loss with useful data aggregation.

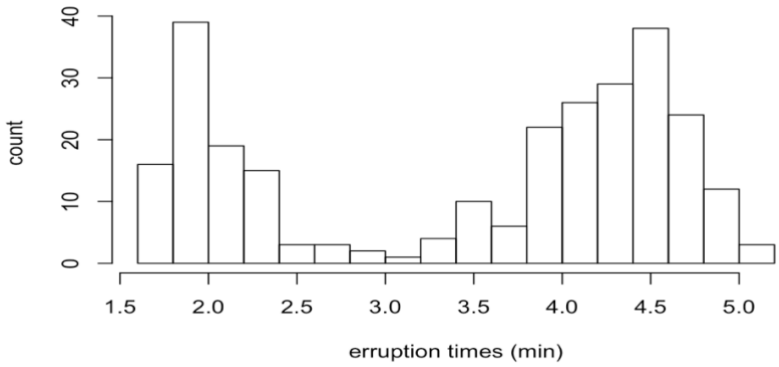


Figure 2 - Histogram of the eruption times of the Old Faithful geyser

We notice that the Old Faithful geyser has two typical eruption times which confirms what we have seen earlier in the strip plot.

- Line plots: A line plot is a graph that displays a relationship between x and y as a line in a Cartesian coordinate system. The relations may correspond to an abstract mathematical function or to a relation between two samples (for example : dataframe columns).

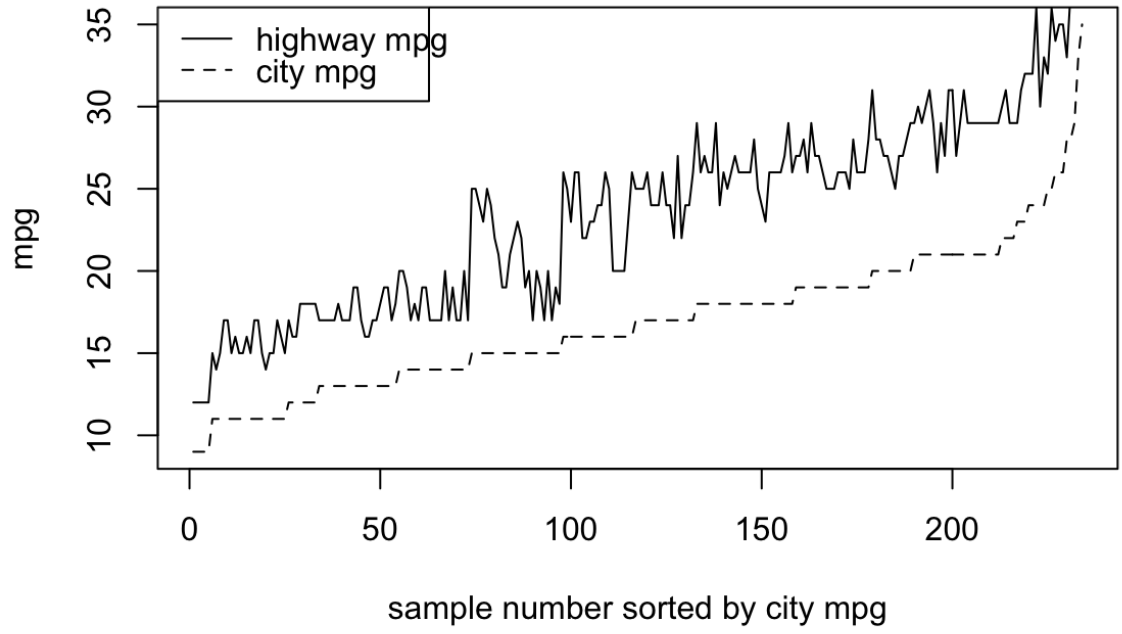


Figure 3 - Line plots

We see here two line plots. The top line plot is the highway mpg and the bottom line plot is city mpg from the *mtcars* dataset in R. We notice that as the city mpg increases, the highway mpg tend to increase as well. Line plots are very useful because we can retrieve critical information.

- Scatter Plots: It graphs the relationships between two numeric variables. It graphs each pair of variables as a point in a two dimensional space whose coordinates are the corresponding x and y values.

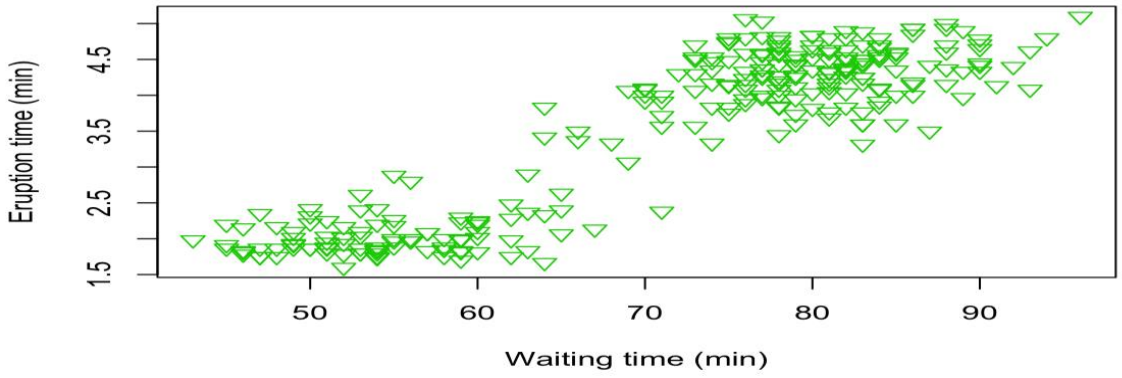


Figure 4 - Scatter plot reflecting the relationship between the eruption time and waiting time

Here we plot the scatter plot of the eruption time and the waiting time of the Old Faithful geyser. We notice again a similar situation where we have two clusters. The first cluster have a short wait time and a short eruption time whereas the second cluster have a short wait time and a short eruption time. This make sense, because if the gazer takes longer time before it erupts, meaning the wait time is longer, more pressure builds up and the eruption is longer.

- Multivariate Scatter Plots: These plots are designed to reveal the relationship among several variables simultaneously. In this example we are going to change the marker size to reveal a relationship between three different numeric variables in a scatter plot. Because scatter plot are two-dimensional, we need to use the marker size to reflect the third variable.

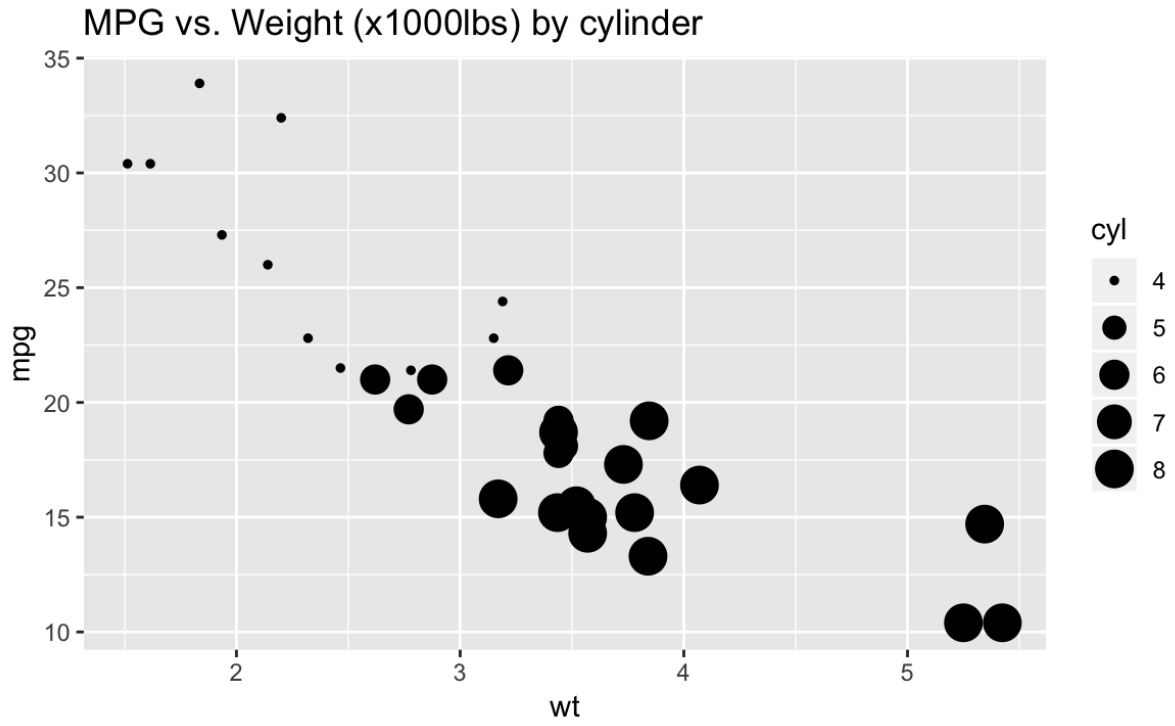


Figure 5 - Multivariate Scatter plot of the mpg ~ weight ~ cylinder

In the figure above, we used the dataset *mpg* which looks similar to the *mtcars* dataset but is larger and have more recent data.

We can see that there is an inverse relationship between the weight variable and the miles per gallon variable. In fact heavier cars tends to have a lower miles per gallon. We also see, the heavier cars tends to have more cylinders.

- Box plots: It's an alternative to histograms that are usually more lossy, in the sense that they lose more data. But they emphasize quantiles and outliers in a way a histogram cannot. Sometimes a histogram is more useful, but in other cases, boxplots are more useful and they reveal information that a histogram does not have.

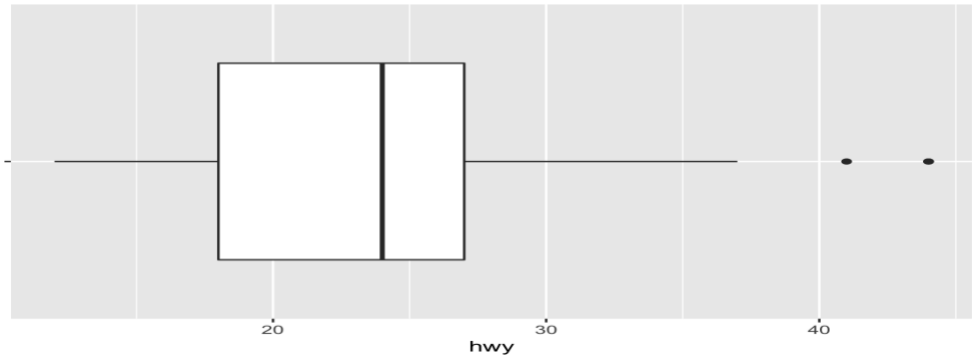


Figure 6 - Box plot of the highway variable

We conclude from this graph that the median highway mpg is around 24, with the central 50% of the data falling within the box that spans the range from 18 to 27. There are two high outliers over 40, but otherwise the remaining data lie within the whiskers between 12 and 37. The fact that the median line is right of the middle of the box hints that the distribution is skewed to the right.

It is sometimes convenient to plot several box plots side by side in order to compare data corresponding to different values of a factor variable. We demonstrate this by graphing below box plots of highway mpg for different classes of vehicles. Note that we re-order the factors of the class variable in order to sort the box plots in order of increasing highway mpg medians. This makes it easier to compare the different populations.

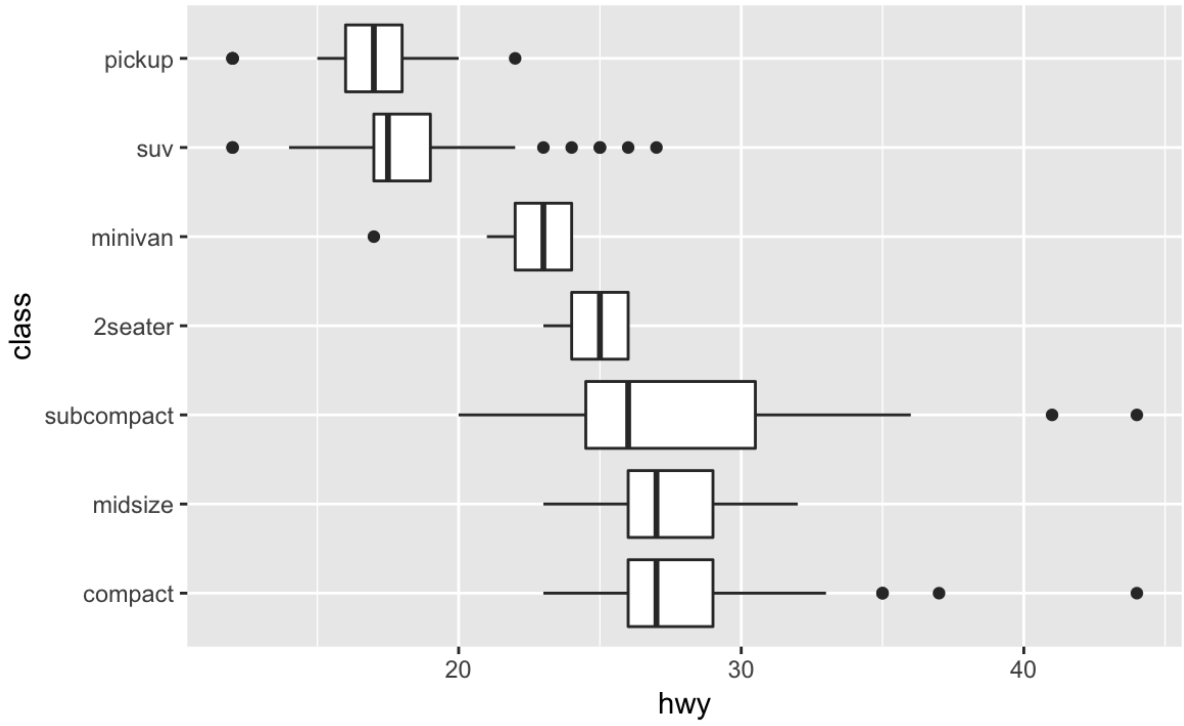


Figure 7 - Box plots of all the variable ordered of highway mpg medians in the mpg dataset

Data Processing

Data processing is an important step in Machine Learning. Data gathering methods are often loosely controlled, resulting in out-of-range values, impossible data combinations, missing values, etc. Analyzing the data that has not been carefully screened for such problems can produce misleading results. Thus, the representation and the quality of the data is first and foremost before running an analysis.
In this section we will see how to study three different situations:

- **Missing data**: Some of the measurements are not available due to data corruption or difficulty in obtaining the data.
- **Outliers**: Some of the measurements are highly atypical of the data distribution.
- **Skewed data**: The data is highly skewed making its visualization and analysis difficult.

Missing Data:

Data may be missing for a variety of reasons. Perhaps it was corrupted during its transfer or storage. Perhaps some instances in the data collection process were skipped due to difficulty or price associated with obtaining the data. Or perhaps the data was simply unavailable for some other reason.

Denoting n data instances by $x^{\{i\}}, i = 1, \dots, n$ (corresponding for example to dataframe rows), missing data implies that for each $i = 1, \dots, n$ we have a set $A_i \subset \{1, \dots, n\}$ of indices for which the measurements are missing (A may potentially be the empty set in which case no data is missing). In other words, $x_j^{(i)}$ (the j variable of the i sample) is missing if $j \in A_i$.

If the probability of an observation being missing does not depend on observed or unobserved measurements, we say that it is missing completely at random (MCAR). For example, in the case of users rating movies using 1-5 stars, we consider ratings of specific movies as data frame columns and ratings associated with specific users as data frame rows. Since some movies are more popular than others, the probability of missingness depends on the movie title as well as the movie rating, which violates the MCAR definition.
A more relaxed concept is data missing at random (MAR). This occurs when given the observed data, the probability that data is missing does not depend on the unobserved data. Consider, for example, a survey recording gender, race, and income. Out of the three questions, gender and race are not very objectionable questions, so we assume for now that the survey respondents answer these questions fully. The income question is more sensitive and users may choose to not respond to for privacy reasons. The tendency to report income or to not report income typically varies from person to person. If it only depends on gender and race, then the data is MAR. If the decision whether to report income or not depends also on other variables that are not in the dataframe (such as age or profession), the data is not MAR.

Some data analysis techniques are specifically designed to allow for missing data. In general, however, most methods are designed to work with fully observed data. Below are some general ways to convert missing data to non-missing data.

- Remove all data instances (for example data rows) containing missing values.
- Replace all missing entries with a substitute value, for example the mean of the observed instances of the missing variable.
- Estimate a probability model for the missing variable and replace the missing value with one or more samples from that probability model.

In the case of MCAR, all three techniques above are reasonable in that they may not introduce systematic errors. In the more likely case of MAR or non-MAR data, the models above may introduce systematic bias into the data analysis process.

Outliers:

There are two different definitions for outliers. This first considers outliers as corrupted values. That is the case, for example, with human errors during a manual process of entering measurements in a spreadsheet. The second definition considers outliers to be non-corrupt values, but nevertheless are substantially unlikely given our modeling assumptions.
Data analysis based on outliers may result in drastically wrong conclusions. This is pretty clear in the case of corrupted outliers. But it may also be the case with the second definition of outliers, especially when the model used in the data analysis does not account for the extreme observations.

Some models are sensitive to outliers and building such models based on data with outliers can lead to drastically inaccurate predictions. On the other hand, removing outliers is tricky as the resulting model may conclude that future outliers are unlikely to occur.

Robustness describes a lack of sensitivity of data analysis procedures to outliers. An example for a non-robust procedure is computing the mean of n numbers. Assuming a symmetric distribution of samples around 0, we expect the mean to be zero, or at least close to it. But, the presence of a single outlier (very positive value or very negative value) may substantially affect the mean calculation and drive it far away from zero, even for large n.

An example for a robust data analysis procedure is the median, which will not be affected by a single outlier even if it has extreme values. We illustrate this with a hypothetical data $n+1$ values $\{ x^{(1)}, x^{(2)}, \dots, x^{(n)}, e \}$, with a median b and a mean a. The value $e > \max(x^{(1)}, \dots, x^{(n)})$ is a single high value outlier. Fixing the n observations and increasing e to infinity the median will remain constant at b, while the mean would grow to infinity. This happens regardless of the value of n. In other words, no matter the size of the dataset, a single extreme outlier will affect the mean in a substantial way but it will not affect the median.

Here are three popular techniques for dealing with outliers:

- **Truncating**: Remove all values deemed as outliers.
- **Winsorization**: Shrink outliers to border of main part of data. One special case of this is to replace outliers with the most extreme of the remaining values.
- **Robustness**: Analyze the data using a robust procedure.

Data Transformations

In many cases, data is drawn from a highly-skewed distribution that is not well described by one of the common statistical distributions. In some of these cases, a simple transformation may map the data to a form that is well described by common distributions, such as Gaussian or Gamma distribution. A suitable model can then be fitted to the transformed data (if necessary, predictions can be made on the original scale by inverting the transformation).

Power transformations are a family of data transformations for non-negative values (parameterized by $\lambda \in \mathbb{R}$), defined as follows.

$$f_\lambda(x) = \begin{cases} \dfrac{(x^\lambda - 1)}{\lambda}, & \lambda > 0 \\ \log x, & \lambda = 0 \\ -\dfrac{x^\lambda - 1}{\lambda}, & \lambda < 0 \end{cases} \qquad x > 0, \quad \lambda \in \mathbb{R}.$$

The reason for the algebraic form $\frac{(x^\lambda - 1)}{\lambda}$ rather than the simpler x^λ is that the former choice makes $f_\lambda(x)$ continuous in λ as well as in x. The minus sign in the last case ensures that the transformation does not re-order the data points (taking negative powers reverses ordering).
The power transformations can also be used to transform negative data by adding a number large enough so all values are non-negative and then proceeding according to the definition above.
Intuitively, the power transform maps x to x^λ, up to multiplication by a constant and addition of a constant. This mapping is convex for $\lambda > 1$ and concave for $\lambda < 1$. A choice of $\lambda < 1$ removes right-skewness (data has a heavy tail to the right) with smaller values of λ resulting in a more aggressive removal of skewness. Similarly, a choice of $\lambda > 1$ removes left-skewness.
One way to select this parameter λ is to try different values, graph the resulting histograms, and select one of them. There are also more sophisticated methods for selecting λ based on the maximum likelihood method.

Logistic Regression

Logistic regression is the appropriate regression analysis to conduct when the dependent variable is binary. Like all regression analysis, the logistic regression is a predictive analysis. Logistic regression is used to describe data and to explain the relationship

between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.

Binary Classification

Binary classification is the most basic task in machine learning, and yet the most frequent. Binary classifiers often serve as the foundation for many high tech ML applications such as ad placement, feed ranking, spam filtering, and recommendation systems.

Below we assume that x is a d-dimensional vector $x = (x_1, \dots, x_d)$ of real numbers, and y is a scalar denoting the label: $+1$ or -1. We will denote by θ the vector of parameters defining the classifier, whose dimensionality is the same as that of x: $\theta = (\theta_1, \dots, \theta_d)$. The inner product between θ. The inner product between θ and x is defined as follows:

$$\langle x, \theta \rangle = \theta_1 x_1 + \cdots + \theta_d x_d$$

The binary classification task is defined as follows: Given a vector of features x, assign a label y of $+1$ or -1. A binary classifier is a rule that makes such mapping for arbitrary vector x. The classifier is typically learned based on training data composed of n labeled vectors: $\left(x^{(1)}, y^{(1)}\right), \dots, \left(x^{(n)}, y^{(n)}\right)$. Note that each $x^{(i)}$ is a vector and $y^{(i)}$ its $+1$ or -1 label.

For example, in the case of the LinkedIn feed, x can contain measurements characterizing the user and item, and y represents whether the user clicks or not on the item. An example for a feature could be $x_5 = 1$ if the user has more than 500 connections and the item is a connection update. There could be thousands or more such features. The training data is a sequence of user-item and click/no-click information, and at serving time the classifier tries to predict whether the user will click on the item feed or not.

Linear Classifiers and Hyperplanes

A linear classifier is a classifier that has the following algebraic form: $y = sign(\langle \theta, x \rangle)$. That is, the predicted label of the vector x is the sign of the inner product of that vector with another vector θ that is called the parameter vector of the classifier. If $\langle x, \theta \rangle$ is positive the predicted class is $+1$ and if it is negative the predicted class is -1.

Despite their simplicity (and probably because of it), linear classifiers are the most widely used. There are several reasons:

a) They are easy to train
b) They can predict labels very fast at serve time
c) We know quite well the statistical theory of linear classifiers leading to effective modeling strategies.

Arguably, the most widely used linear classifiers is logistic regression. Logistic regression, or its variations, power main functions in big companies like Google, LinkedIn and Amazon, as well as in small startups.

Probabilistic Classifiers and Maximum Likelihood

The above definition of classifier defines a map from a vector of features x to a $+1$ or -1 label. It is useful to have also a measure of confidence of that prediction as well as a way to interpret that confidence in an objective manner. Probabilistic classifiers provide that tool by defining the probabilities of the labels $+1$ and -1 given the feature vector x. Recall the fact that the two probabilities must be non-negative and sum to one. The probabilities of the two labels given a feature vector are written as $p_\theta(Y = 1 \mid X = x)$ and $p_\theta(Y = -1 \mid X = x)$ where θ is a parameter vector that defines the classifier.
Statistic theory strongly suggests that probabilistic classifiers should be learned from the labeled vectors $\left(x^{(1)}, y^{(1)}\right), \ldots, (x^{(n)}, \ldots, y^{(n)})$ using the maximum likelihood estimator (MLE), defined below:

$$\hat{\theta}_{MLE} = \arg\max p_\theta\left(Y = y^{(1)} \mid X = x^{(1)}\right) \ldots\ p_\theta\left(Y = y^{(n)} \mid X = x^{(n)}\right)$$

The MLE attempts to find the parameter values that maximize the likelihood function $\mathcal{L}(\theta; x)$, given the observations.

High Dimensions, Overfitting, and Regularization

When x is high dimensional the issue of overfitting comes up: there are too many parameters to estimate from the available labeled data, resulting in a trained classifier that fits random noise patterns that exist in the data, rather than the rule governing the likely behavior. For example, consider an extreme case (for illustration purposes) of $d = 10^6$ and $n = 2$. Obviously we cannot properly learn a million parameters and how they define the linear decision boundary in a million dimensional space based on 2 labeled data vectors.
These situations should be handled carefully. We will see different regularization techniques later on.

Linear Regression

Linear regression is used to predict the value of an outcome variable Y based on one or more input predictor variables X. The aim is to establish a linear relationship between the predictor variable(s) and the response variable, so that, we can use this formula to estimate the value of the response Y, when only the predictors values are known.

The aim of linear regression is to model a continuous variable Y as a mathematical function of one or more X variable(s), so that we can use this regression model to predict the Y when only the X is known. This mathematical equation can be generalized as follows:

$$Y = \beta_1 + B_2 X + \epsilon$$

where, β_1 is the intercept and β_2 is the slope. Collectively, they are called regression coefficients. ϵ is the error term, the part of Y the regression model is unable to explain.

Example Problem

For this analysis, we will use the *cars* dataset that comes with R by default. It consists of 50 observations and 2 variables, *dist* and *speed*. Before we begin building the regression model, it is a good practice to analyze and understand the variables.

Graphical Analysis

We will try to build a simple regression model that we can use to predict the distance by establishing a statistically significant linear relationship with the speed. We will try to graph different types of plots to visualize different behaviors:

1. Scatter plot: Visualize the linear relationship between the predictor and response
2. Box plot: To spot any outlier observations in the variable. Having outliers in your predictor can drastically affect the predictions as they can easily affect the direction/slope of the limit of best fit.
3. Density plot: To see the distribution of the predictor variable. Ideally, a close to normal distribution, without being skewed to the left or right is preferred.

Scatter plot

Scatter plots can help visualize any linear relationships between the dependent (response) variable and independent (predictor) variables. Ideally, if you are having multiple predictor variables, a scatter plot is drawn for each one of them against the response, along with the line of best as seen below.

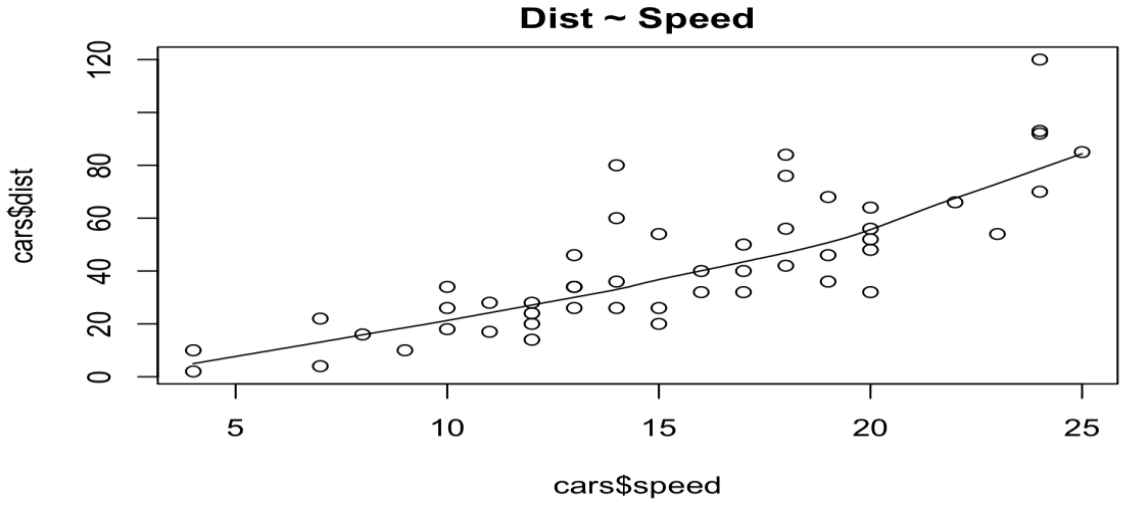


Figure 8 - Scatter plot between the speed and the distance variable

The scatter plot along with the smoothing line above suggests a linearly increasing relationship between the 'dist' and 'speed' variables. This is a good thing, because, one of the underlying assumptions in linear regression is that the relationship between the response and predictor variables is linear and additive.

Box plot – Check for outliers

Generally, any datapoint that lies outside the 1.5 * interquartile-range (IQR) is considered an outlier, where the IQR is calculated as the distance between the 25th percentile and 75th percentile values for that variable.

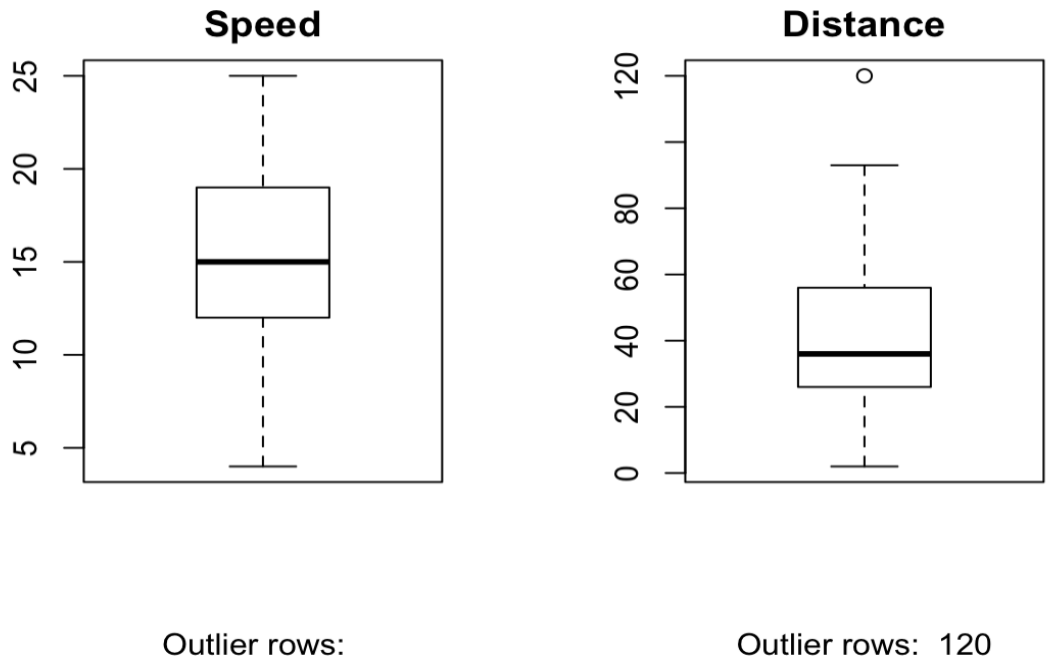


Figure 9 - Box plot of speed and distance variable.

We notice that the distance has an outlier.

Density plot–Check if the response variable is close to normality

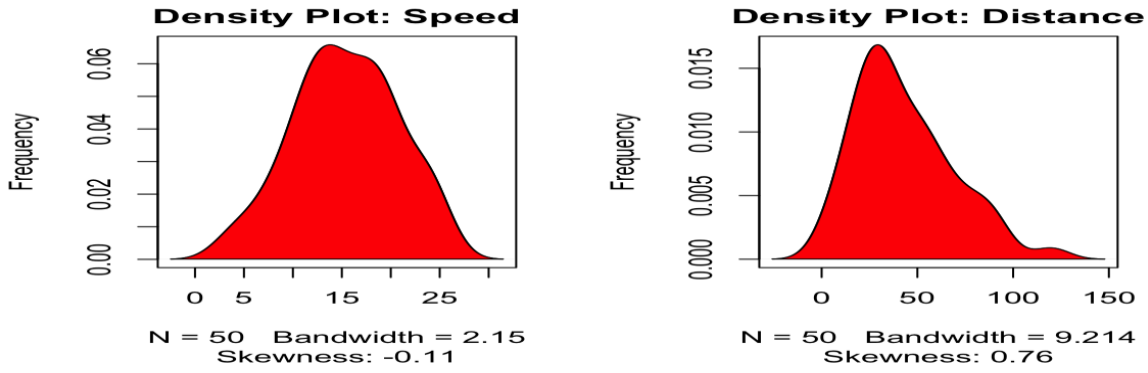


Figure 10 - Density plots of the speed and distance variable

We notice that both of the variables are close to normality with a few left skewness for the distance.

Correlation

Correlation is a statistical measure that suggests the level of linear dependence between two variables, that occur in pair – just like what we have here in speed and dist. Correlation can take values between -1 to +1. If we observe for every instance where speed increases, the distance also increases along with it, then there is a high positive correlation between them and therefore the correlation between them will be closer to 1. The opposite is true for an inverse relationship, in which case, the correlation between the variables will be close to -1.

A value closer to 0 suggests a weak relationship between the variables. A low correlation (-0.2 < x < 0.2) probably suggests that much of variation of the response variable (Y) is unexplained by the predictor (X), in which case, we should probably look for better explanatory variables.

Build a Linear Model

Now that we have seen the linear relationship pictorially in the scatter plot and by computing the correlation, let's see the syntax for building the linear model. The function used for building linear models is lm(). The lm() function takes in two main arguments, namely: 1. Formula 2. Data. The data is typically a data.frame and the formula is a object of class formula. But the most common convention is to write out the formula directly in place of the argument as written below:

```
linearModel = lm(dist~speed, data=cars) #build linear regression model on full data
print(linearModel)
```

```
##
## Call:
## lm(formula = dist ~ speed, data = cars)
##
## Coefficients:
## (Intercept)       speed
##    -17.579       3.932
```

We have $\text{dist} = -17.579 + 3.932 * \text{speed}$.

Linear Regression Diagnostics

Now the linear model is built and we have a formula that we can use to predict the *dist* value if a corresponding speed is known. Is this enough to actually use this model ? No! Before using a regression model, we have to ensure that it is it statistically significant. How do you ensure this? Let's begin by printing the summary statistics for the linear model.

```
summary(linearModel)
```

```
##
## Call:
## lm(formula = dist ~ speed, data = cars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -29.069  -9.525  -2.272   9.215  43.201
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -17.5791     6.7584  -2.601   0.0123 *
## speed         3.9324     0.4155   9.464 1.49e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.38 on 48 degrees of freedom
## Multiple R-squared:  0.6511, Adjusted R-squared:  0.6438
## F-statistic: 89.57 on 1 and 48 DF,  p-value: 1.49e-12
```

The summary statistics above tells us a number of things. One of them is the model p-Value and the p-Value of individual predictor variables The p-Values are very important because, We can consider a linear model to be statistically significant only when both these p-Values are less that the pre-determined statistical significance level, which is ideally 0.05. This is visually interpreted by the significance stars at the end of the row. The more the stars beside the variable's p-Value, the more significant the variable.

When the model coefficients and standard error are known, the formula for calculating t Statistic and the p-value is as follows:

$$tStatistic = \frac{\beta - coefficient}{Std.Error}$$

R-Squared and Adjusted R-Squared

What R-Squared tells us is the proportion of variation in the dependent variable that has been explained by the model.

$$R^2 = 1 - \frac{SSE}{SST}$$

where, SSE is the sum of squared errors given by $SSE = \sum_i^n (y_i - \hat{y}_i)^2$ and $SST = \sum_i^n (y_i - \bar{y}_i)^2$ is the sum of squared total. Here, $\hat{y}_i$ is the fitted value for observation i and $\bar{y}$ is the mean of Y. We don't necessarily discard a model based on a low R-Squared value. It's a better practice to look at the AIC and prediction accuracy on validation sample when deciding on the efficiency of a model.

Now that's about R-Squared. What about adjusted R-Squared? As you add more X variables to your model, the R-Squared value of the new bigger model will always be greater than that of the smaller subset. This is because, since all the variables in the original model is also present, their contribution to explain the dependent variable will be present in the super-set as well, therefore, whatever new variable we add can only add (if not significantly) to the variation that was already explained. It is here, the adjusted R-Squared value comes to help. Adjusted R-Squared penalizes total value for the number of terms (read predictors) in your model. Therefore, when comparing nested models, it is a good practice to look at adjusted R-squared value over R-squared.

$$R^2_{adj} = 1 - \frac{MSE}{MST}$$

where, MSE is the $mean\ squared\ error$ given by $MSE = \frac{SSE}{n-q}$ and $MST = \frac{SST}{n-1}$ is the $mean\ squared\ total$, where n is the number of observations and q is the number of coefficients in the model. Therefore, by moving around the numerators and denominators, the relationship between R^2 and R^2_{adj} becomes:

$$R^2_{adj} = 1 - \frac{(1 - R^2)(n-1)}{(n-q)}$$

Standard Error and F-Statistic

Both standard error and F-Statistic are measures of goodness of fit.

$$Std.Error = \sqrt{MSE} = \sqrt{\frac{SSE}{n-q}}$$

$$F - Statistic = \frac{MSR}{MSE}$$

where, n is the number of observations, q is the number of coefficients and MSR is the $mean\ square\ regression$, calculated as,

$$MSR = \frac{\sum_i^n (\hat{y}_i - \bar{y}_i)}{q-1} = \frac{SST - SSE}{q-1}$$

AIC and BIC

The **Akaike's information criterion - AIC** and the **Bayesian Information criterion - BIC** are measures of the goodness of fit of an estimated statistic model and can also be used for model selection.
For model comparison, the model with the lowest AIC and BIC score is preferred.

How to know if the model is best fit for your data?

The most common metrics to look at while selecting the model are:

Statistic	Criterion
R-Squared	Higher the better (> 0.70)
Adj R-Squared	Higher the better
F-Statistic	Higher the better
Std.Error	Close to 0 the better
t-Statistic	Should be greater 1.96 for p-value to be less than 0.05
AIC	Lower the better
BIC	Lower the better

Predicting Linear Models

So far we have seen how to build a linear regression model using the whole dataset. If we build it that way, there is no way to tell how the model will perform with new data. So the preferred practice is to split your dataset into a 80:20 sample (training:test), then, build the model on the 80% sample and then use the model thus built to predict the dependent variable on test data.

Step 1: Create the training (development) and test (validation) data samples from original data.

```
# Create Training and Test data -
set.seed(100)  # setting seed to reproduce results of random sampling
trainingRowIndex <- sample(1:nrow(cars), 0.8*nrow(cars))  # row indices for training data
trainingData <- cars[trainingRowIndex, ]  # model training data
testData  <- cars[-trainingRowIndex, ]  # test data
```

Step 2: Develop the model on the training data and use it to predict the distance on test data

```
# Build the model on training data -
lmMod <- lm(dist ~ speed, data=trainingData)  # build the model
distPred <- predict(lmMod, testData)  # predict distance
```

Step 3: Review diagnostic measures

```
summary(lmMod)
##
## Call:
## lm(formula = dist ~ speed, data = trainingData)
##
## Residuals:
##    Min     1Q  Median     3Q    Max
## -23.350 -10.771  -2.137   9.255  42.231
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -22.657      7.999  -2.833  0.00735 **
## speed          4.316      0.487   8.863 8.73e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.84 on 38 degrees of freedom
## Multiple R-squared:  0.674,  Adjusted R-squared:  0.6654
## F-statistic: 78.56 on 1 and 38 DF,  p-value: 8.734e-11
```

From the model summary, the model p value and predictor's p value are less than the significance level, so we know we have a statistically significant model. Also, the R-Squared and Adjusted R-Squared are comparative to the original model built on full data.

Step 4: Calculate prediction accuracy and error rates

A simple correlation between the actuals and predicted values can be used as a form of accuracy measure. A higher correlation accuracy implies that the actuals and predicted values have similar directional movement, i.e. when the actuals values increase the predicteds also increase and vice-versa.

```
actula_preds = data.frame(cbind(actuals=testData$dist, predicteds=distPred)) # make actuals_predicteds dataframe.
correlation_accuracy = cor(actula_preds)
head(actula_preds)
```

```
##    actuals predicteds
## 1        2  -5.392776
## 4       22   7.555787
## 8       26  20.504349
## 20      26  37.769100
## 26      54  42.085287
## 31      50  50.717663
```

```
correlation_accuracy
```

```
##             actuals predicteds
## actuals    1.0000000  0.8277535
## predicteds 0.8277535  1.0000000
```

References

Data Processing, *https://s3.amazonaws.com/content.udacity-data.com/courses/gt-cse6242/recommended+reading/processing.pdf*

Data Visualization, https://s3.amazonaws.com/content.udacity-data.com/courses/gt-cse6242/recommended+reading/graphics.pdf

Modèles Linéaires(2010), C.Chouquet, Laboratoire de Statistique et de Probabilités – Université Paul Sabatier – Toulouse
https://www.math.univ-toulouse.fr/~barthe/M1modlin/poly.pdf

Inference in High Dimensions and Regularization, Guy Lebanon
https://s3.amazonaws.com/content.udacity-data.com/courses/gt-cse6242/recommended+reading/regularization.pdf

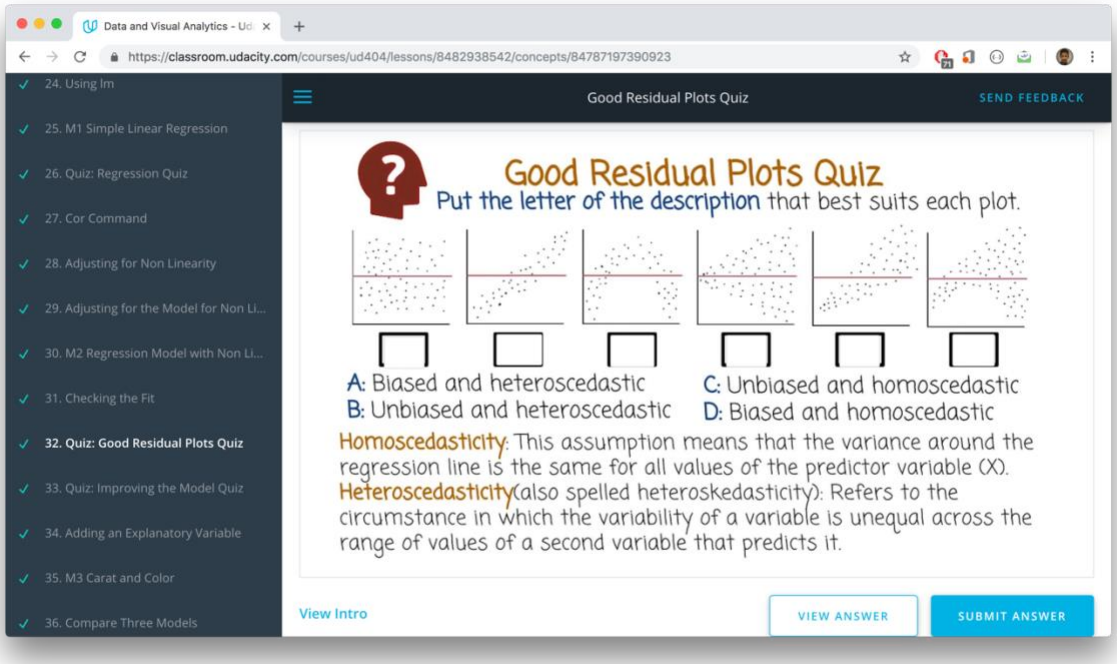
Elegant Graphics for Data Analysis, Hadley Wickham
https://s3.amazonaws.com/content.udacity-data.com/courses/gt-cse6242/recommended+reading/ggplot2-book.pdf

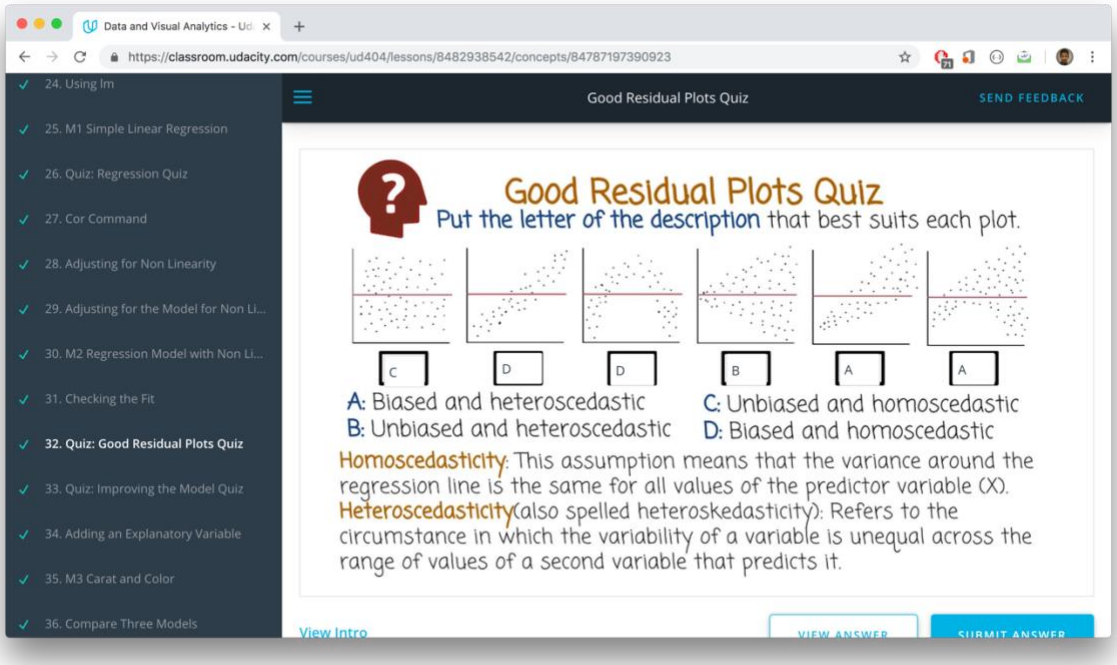
Extending the Linear Model with R, Generalized Linear, Mixed Effects and Nonparametric Regression Models, Julian J. Faraway
https://englianhu.files.wordpress.com/2016/01/faraway-extending-the-linear-model-with-r-e28093-2006.pdf

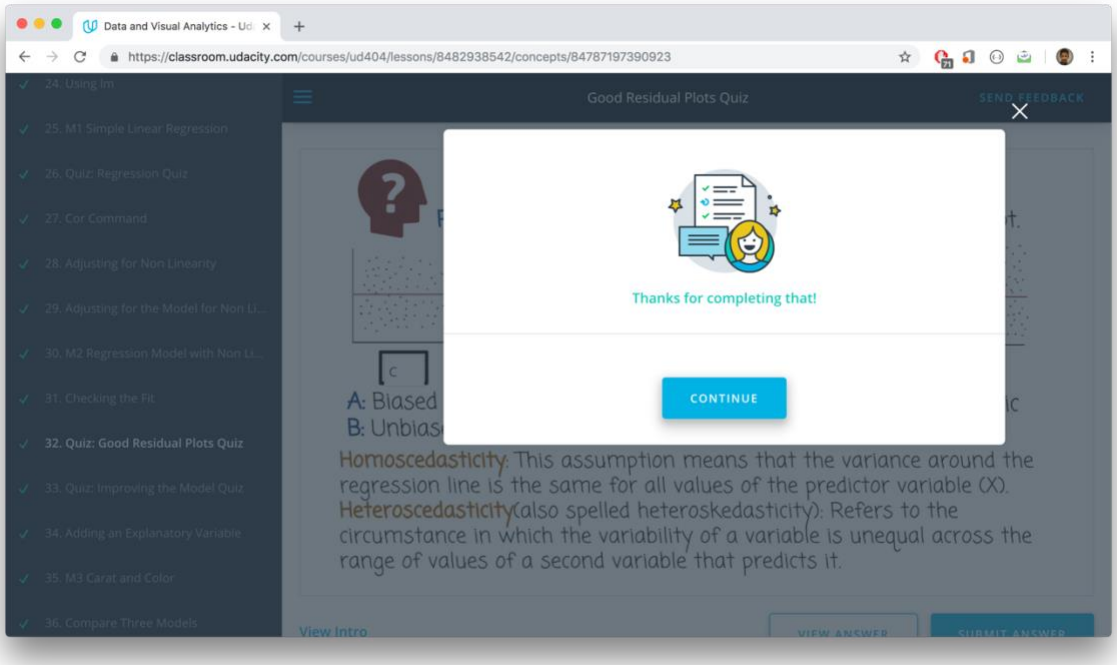
An Introduction to Statistical Learning with Applications in R, Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani
http://www-bcf.usc.edu/~gareth/ISL/ISLR%20First%20Printing.pdf

Annex

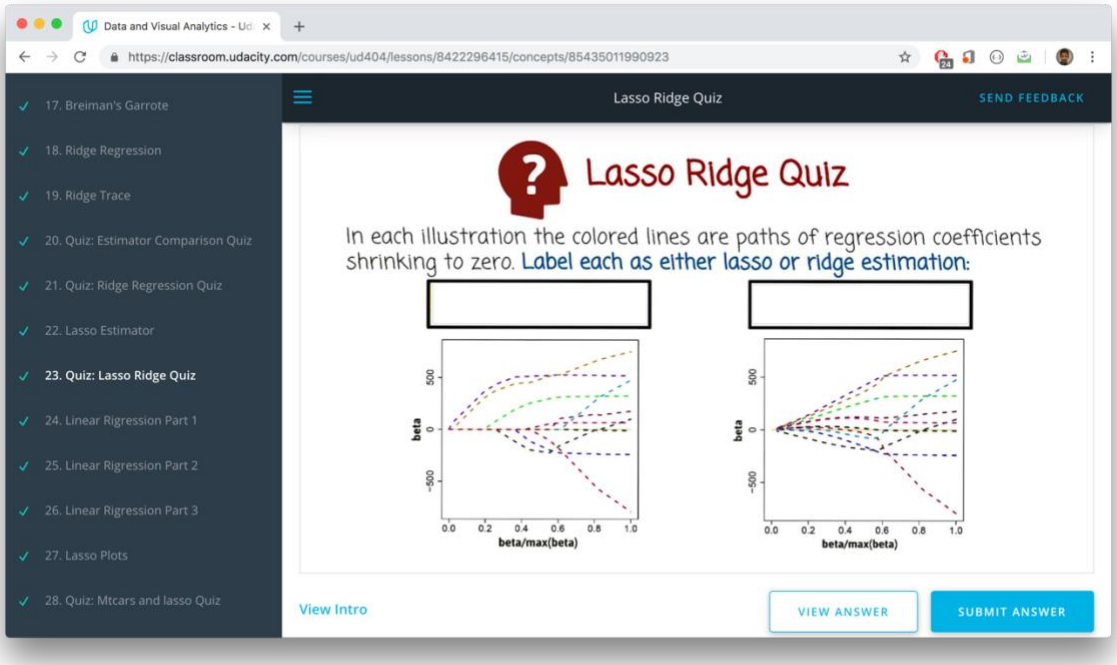
Annex I: Good Residual Plot Quiz

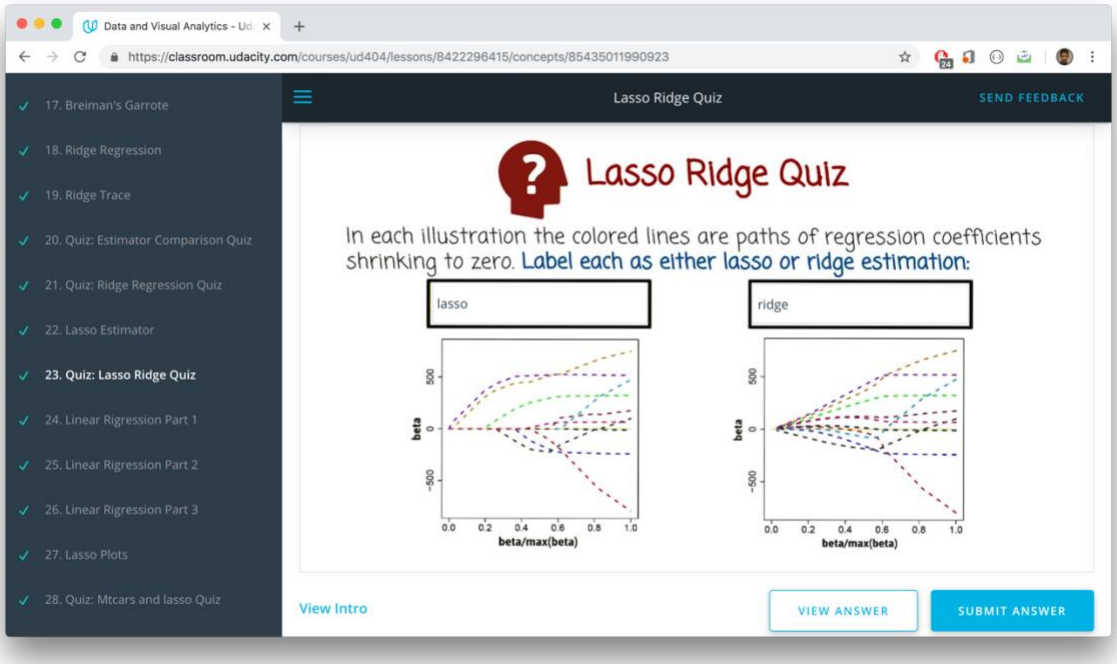


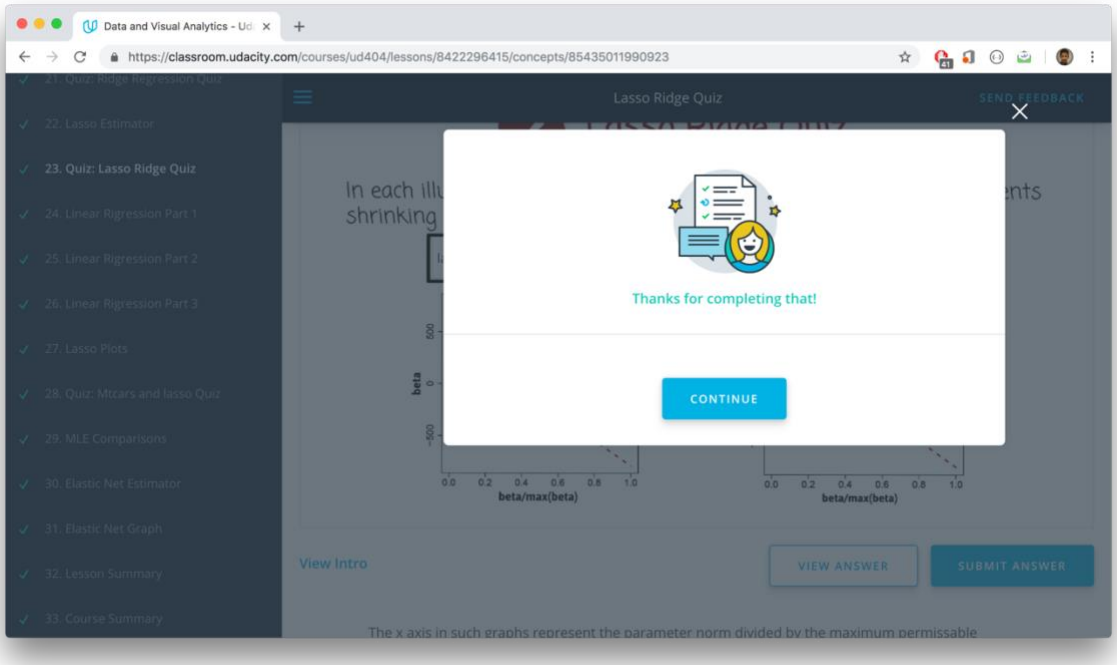




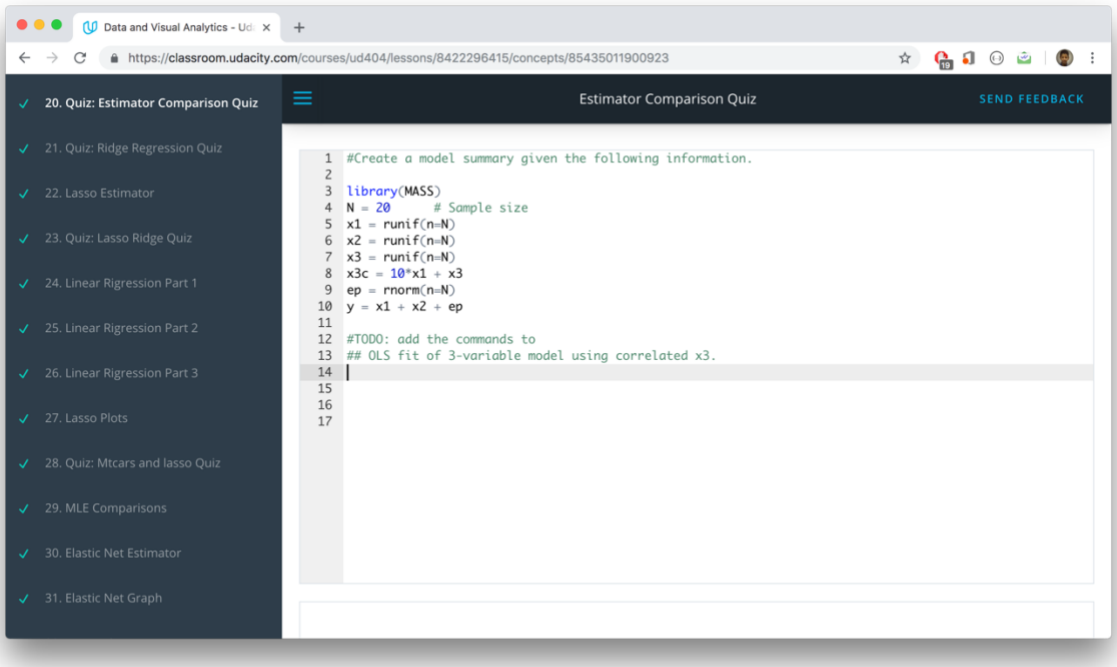
Annex II: Lasso Ridge Quiz

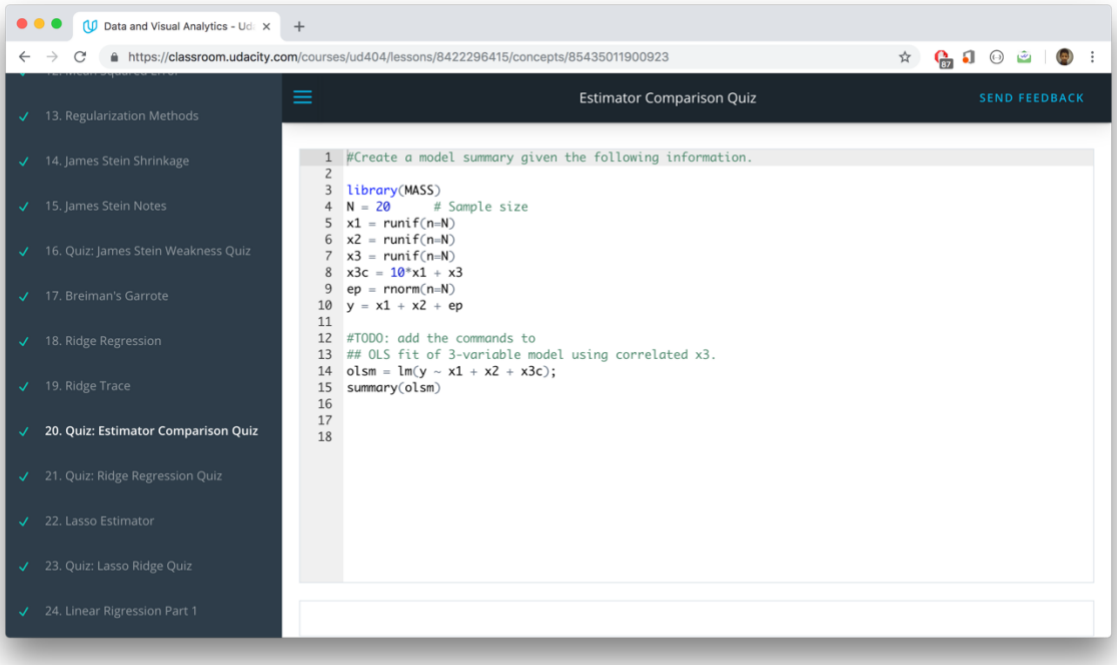


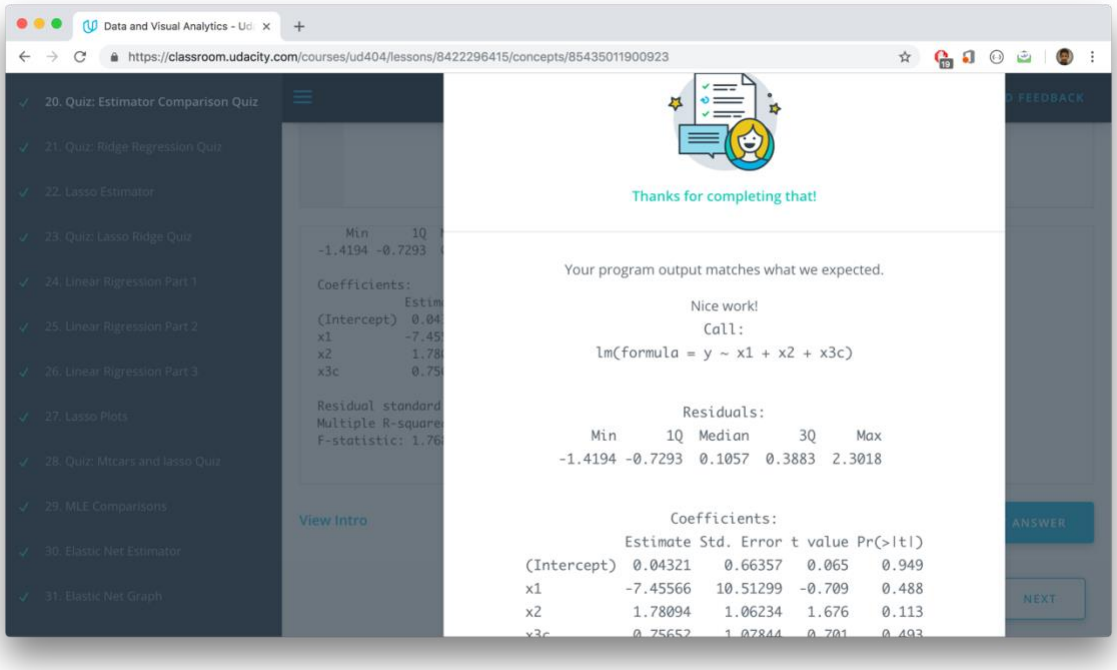

Data and Visual Analytics - Ud
https://classroom.udacity.com/courses/ud404/lessons/8422296415/concepts/85435011990923
17. Breiman's Garrote
18. Ridge Regression
19. Ridge Trace
20. Quiz: Estimator Comparison Quiz
21. Quiz: Ridge Regression Quiz
22. Lasso Estimator
23. Quiz: Lasso Ridge Quiz
24. Linear Rigression Part 1
25. Linear Rigression Part 2
26. Linear Rigression Part 3
27. Lasso Plots
28. Quiz: Mtcars and lasso Quiz
Lasso Ridge Quiz
SEND FEEDBACK
Lasso Ridge Quiz
In each illustration the colored lines are paths of regression coefficients shrinking to zero. Label each as either lasso or ridge estimation:
lasso
ridge
beta
beta/max(beta)
View Intro
VIEW ANSWER
SUBMIT ANSWER


Data and Visual Analytics - Ud
https://classroom.udacity.com/courses/ud404/lessons/8422296415/concepts/85435011990923
22. Lasso Estimator
23. Quiz: Lasso Ridge Quiz
24. Linear Rigression Part 1
25. Linear Rigression Part 2
26. Linear Rigression Part 3
27. Lasso Plots
28. Quiz: Mtcars and lasso Quiz
29. MLE Comparisons
30. Elastic Net Estimator
31. Elastic Net Graph
32. Lesson Summary
33. Course Summary
Lasso Ridge Quiz
SEND FEEDBACK
Thanks for completing that!
CONTINUE
View Intro
VIEW ANSWER
SUBMIT ANSWER
The x axis in such graphs represent the parameter norm divided by the maximum permissable

Annex III: Estimator Comparison Quiz

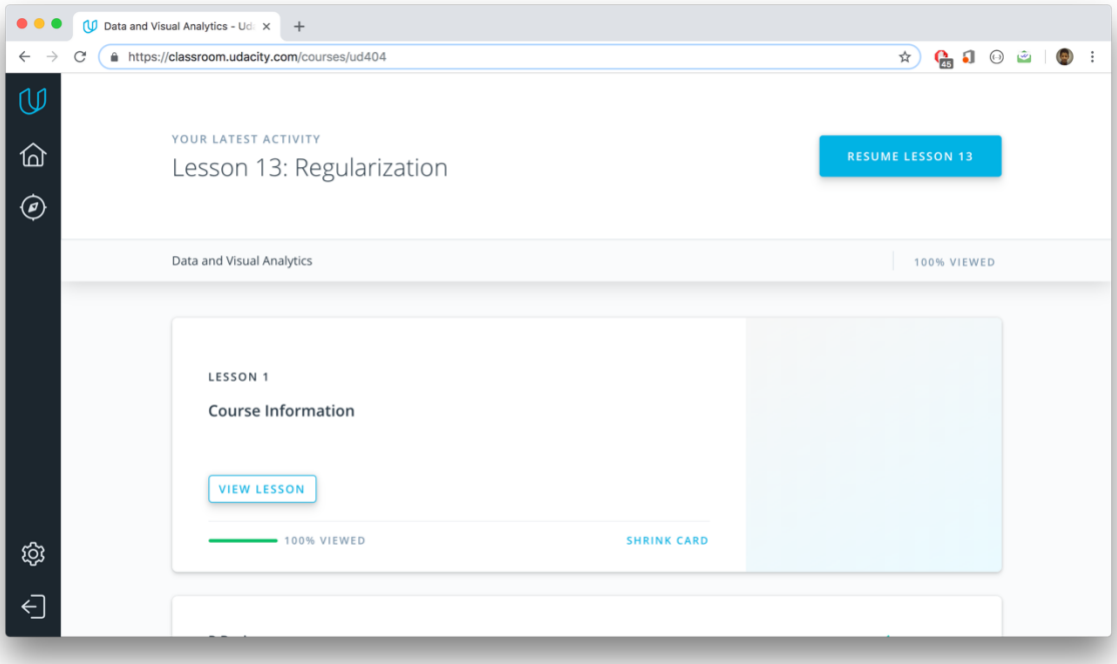


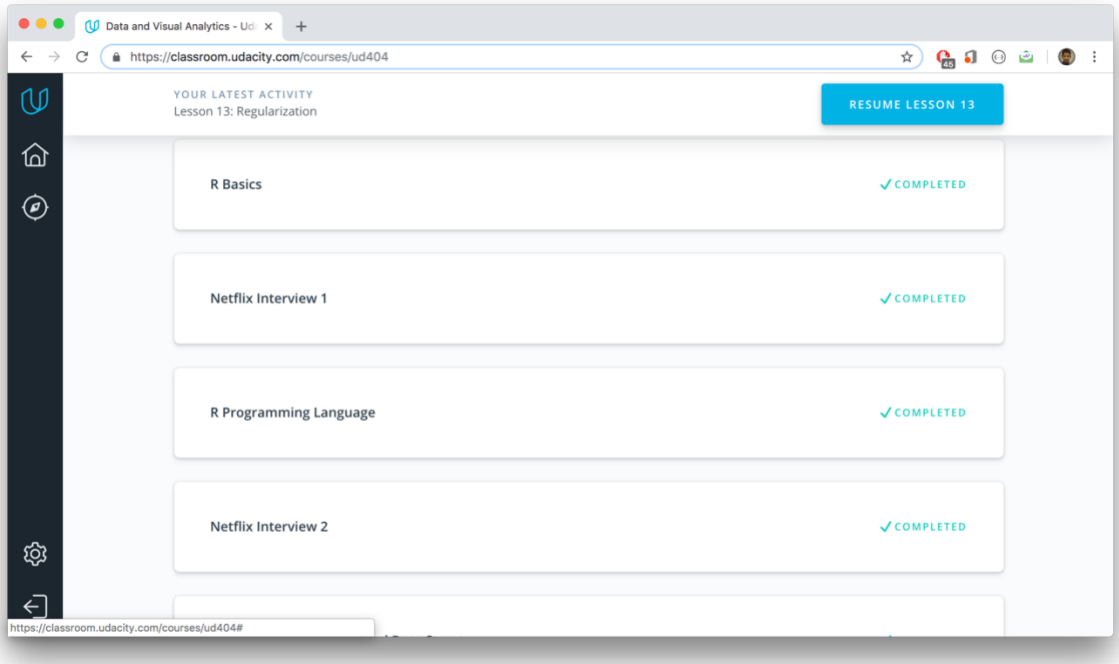


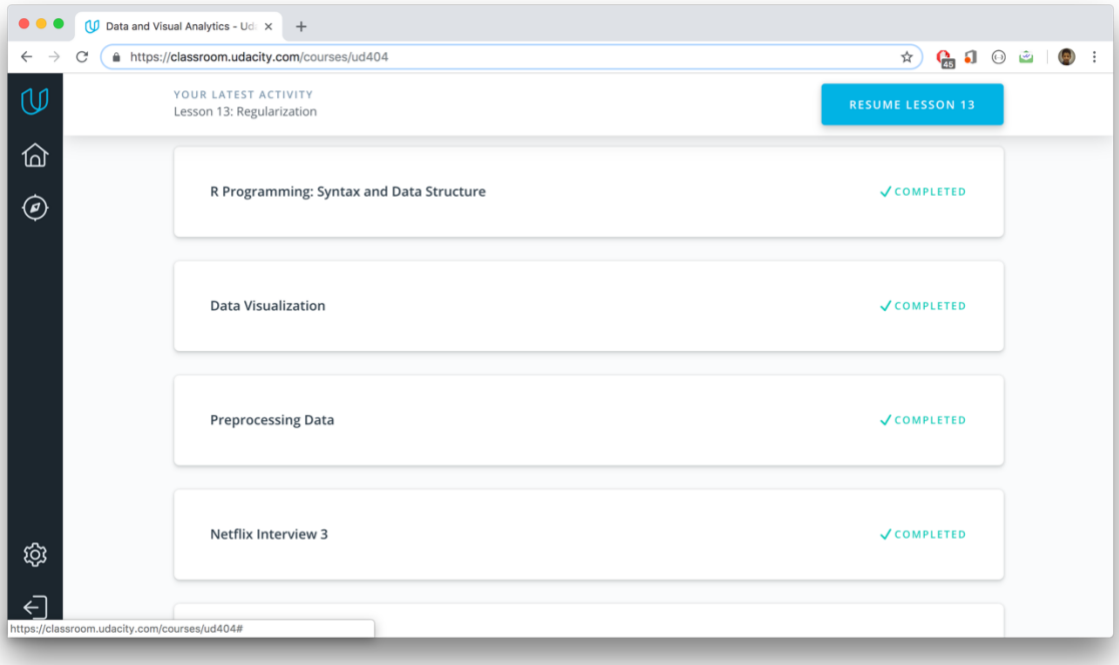


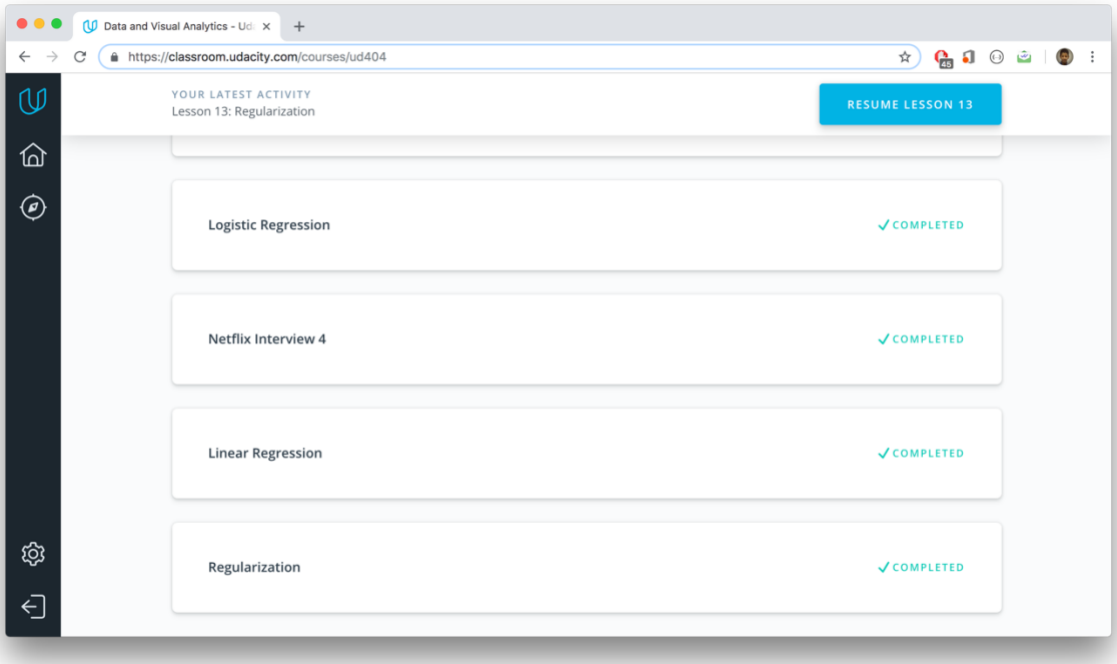
Annex IV: Completion of the MOOC: 100%

Start Date = 20/09/2018
End Date = 09/12/2018




Data and Visual Analytics - Ud
https://classroom.udacity.com/courses/ud404
YOUR LATEST ACTIVITY
Lesson 13: Regularization
RESUME LESSON 13
R Basics
COMPLETED
Netflix Interview 1
COMPLETED
R Programming Language
COMPLETED
Netflix Interview 2
COMPLETED
https://classroom.udacity.com/courses/ud404#


Data and Visual Analytics - Ud
https://classroom.udacity.com/courses/ud404
YOUR LATEST ACTIVITY
Lesson 13: Regularization
RESUME LESSON 13
R Programming: Syntax and Data Structure
COMPLETED
Data Visualization
COMPLETED
Preprocessing Data
COMPLETED
Netflix Interview 3
COMPLETED
https://classroom.udacity.com/courses/ud404#


Data and Visual Analytics - Ud
https://classroom.udacity.com/courses/ud404
YOUR LATEST ACTIVITY
Lesson 13: Regularization
RESUME LESSON 13
Logistic Regression
COMPLETED
Netflix Interview 4
COMPLETED
Linear Regression
COMPLETED
Regularization
COMPLETED