# 25

# *Modernizing the Curricula of Statistics Courses through Statistical Learning*

**Marcus Alexandre Nunes**
*Federal University of Rio Grande do Norte*

## CONTENTS

25.1 Introduction ..... 351
25.2 Relation between Introduction to Big Data Modeling and ASA Guidelines ..... 352
25.2.1 Increased Importance of Data Science ..... 352
25.2.2 Real Applications ..... 353
25.2.3 More Diverse Models and Approaches ..... 354
25.2.4 Ability to Communicate ..... 355
25.3 Case Study ..... 355
25.4 Final Remarks ..... 360
References ..... 360

## 25.1 Introduction

Statistics is an evolving field. The past 20 years have shown how this discipline changed. While new statistical models have appeared, computers became faster and cheaper, allowing professionals to apply these new methods with little trouble. Moreover, more data are generated, collected, and made accessible. Therefore, everyone with Internet access can download new data and analyze it independently. While traditional concepts such as inference, aggregation, likelihood, and experimental design are still valid, there are real-world problems that cannot be addressed in the same fashion as before.

Data visualization, for example, is now much more developed than it was 10 years ago. Consideration of concepts such as color palettes, font options, and different plot types is more commonplace. There are many free statistical software options available for every operational system. New plot types are becoming more popular on a daily basis. Dashboards are simpler to build and update. With this in mind, we believe that universities' undergraduate curricula should be adapted to this new world.

In 2014, the American Statistical Association (ASA) published its guidelines on the undergraduate curriculum of statistics. These guidelines can be summarized in four key points:

1. Increased importance of data science;
2. Real applications;
3. More diverse models and approaches;
4. Ability to communicate.

DOI: 10.1201/9781003261148-30

In this chapter, we report how these guidelines have been applied in a course called *Introduction to Big Data Modeling*, offered since 2015 at the Federal University of Rio Grande do Norte, Brazil. We also report on students' impressions of the discipline as well as some of the tasks that were proposed for them during that time.

*Introduction to Big Data Modeling* is offered regularly as an elective course to second-year students. Its prerequisites are basic statistical inference (*t*-test, ANOVA, simple linear regression) and R programming. It is not a course in which deep mathematical understanding is requested. Few proofs are presented during the course, and the mathematical requirements are equivalent to a statistics 300-level course.

This chapter is structured as follows: in Section 25.2, we show how the course is structured in relation to the ASA Guidelines. In Section 25.3, we present in detail the web-scraping module of the course, showing how we teach our students to collect, process, and prepare real-world data for analysis. We use Section 25.4 to discuss the results of this course and our future plans for it.

## 25.2 Relation between Introduction to Big Data Modeling and ASA Guidelines

### 25.2.1 Increased Importance of Data Science

The impact of data science has been increasing during the past years (Donoho, 2017). One simple Google search for the terms "data science" shows how the interest for this term has increased tenfold during the 2010s, as we can see below (Figure 25.1).

But what is data science? There are many available definitions. We understand data science to be an interdisciplinary field in which statistical and computer science techniques are merged. In our experience, the best results in modeling are obtained by combining these two techniques: the statistical base and the computational power.

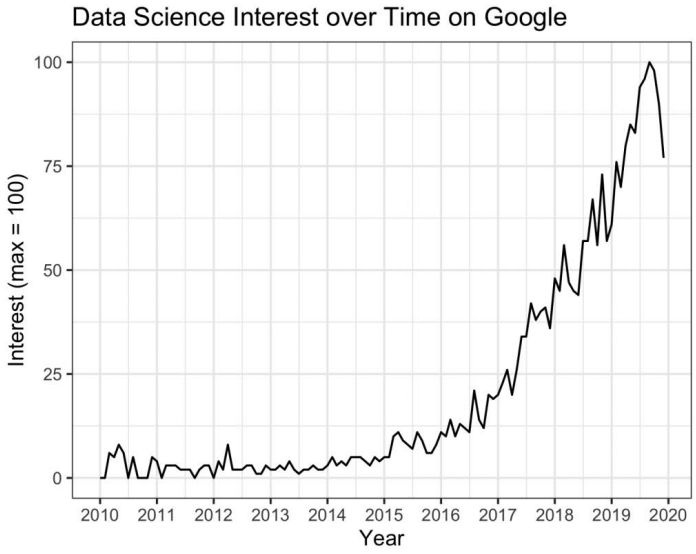


**FIGURE 25.1**
Search volume for "data science" in Google from 2010 to 2020. (Data obtained from Google Trends.)

*Introduction to Big Data Modeling* takes this approach to teach students how to use computational power to solve problems that are not solvable with traditional statistical methods. For example, it is easier to fit a random forest model instead of looking for a parametric distribution that works well with a generalized linear model.

Data science is concerned with data visualization. Many different types of charts are used to generate insights and ideas for analysis, data cleaning, and feature engineering. Our course places great emphasis on data visualization because it is important to show the students how good data visualization is the first step in a successful data analysis.

In addition to the graphical representation of information, this course shows the students how to fit predictive models using machine learning methods. After all, data science is known for the intense use of machine learning and computational resources in the data modeling step. Methods such as cross-validation, random forest, support vector machines, and others demand assets that cannot always be fulfilled by spreadsheets. Therefore, a programming language is needed.

Our programming language of choice is R. While Python is the most popular choice for machine learning modeling, we believe there is no fundamental difference between them. Since R and Python are multiparadigm object-oriented programming languages with many functions written for data science, they have similar performance in this field. We prefer to use R in the course because during the fourth semester, the students have already have been trained in R. Therefore, we can start the data visualization and modeling part at the onset of the course, with no need to teach students how to load data, create loops, or apply functions to objects.

### 25.2.2 Real Applications

One of the pillars of our *Introduction to Big Data Modeling* is the use of real datasets. We understand that the students are more motivated when they see data collected from the real world (Hicks and Irizarry, 2016). We use simple datasets, such as Fisher's Iris dataset, but we also use very complex data, such as the players' attributes from the electronic game FIFA Soccer.

However, there is a spectrum of complexity in the datasets used in this course. One of the first datasets analyzed in the course is Fisher's Iris dataset, a small but interesting collection of flower measurements. Despite its 150 observations in 5 columns, it is possible to run classification and regression algorithms and achieve fast and interesting results. It is interesting to start with this dataset because it does not need to be cleaned or preprocessed, allowing the students to immediately start employing their knowledge in a data science application.

As the course advances, the datasets become more complex. They need to be cleaned, removing observations or grouping them. Categorical variables must be converted to numerical and numerical variables need to be standardized. The students need to learn how to deal with difficult data and we believe the best way to learn the desired skills is by practicing and applying new concepts little by little.

The Internet is full of great sources with interesting datasets. Many federal governments around the world make some of theirs freely available. The US Government's open data can be found at https://www.data.gov/. The Brazilian Institute of Geography and Statistics hosts census results at https://downloads.ibge.gov.br/ (in Portuguese).

There are nongovernmental data sources that provide high-quality data for the classes. Thatmost used in this course is the UCI Machine Learning Repository (https://archive.ics.uci.edu/ml/index.php), maintained by the University of California, Irvine. It has many of the standard datasets used as benchmarks for machine learning algorithms.

More recently, Kaggle (https://www.kaggle.com/datasets) has emerged as one of the most popular places for hosting datasets and machine learning competitions. As of May 2020, Kaggle lists more than 37,000 datasets available for download and free to use.

Besides the datasets provided by the instructor, the students have to work on a project of their own, analyzing a dataset chosen by themselves. The only restriction they have to follow is to use a dataset not analyzed before during the course.

This approach has yielded promising results. When the students are free to choose anything, they tend to focus on subjects in which they have a personal interest. Our students have analyzed data on songs available in Spotify, UFO sightings, evolution of the Human Development Index (HDI) in Brazil in the past three decades, and many other topics.

### 25.2.3 More Diverse Models and Approaches

While many courses in undergraduate level choose to show fewer modeling techniques to the students, proving results and going deep on the math behind them, in our course, we go in the opposite direction. We prefer to present models on a user level, focusing on the model's strengths and limitations. The students are required to intuitively know how the algorithms work, but the majority of the mathematical proofs are omitted. The main result we want from this course is that the students learn when the algorithms work and when they do not.

The list of models taught in the course has changed over time. In its first iteration, one of the classification techniques the students had to learn was linear discriminant analysis (LDA). However, as the course evolved, we decided that the students were working with too many linear classification methods. We changed to random forests, a more general method capable of dealing with nonlinear problems.

*Introduction to Big Data Modeling* is a 15-week course offered during our spring semester. It lasts for one semester, but that has proven to be sufficient to show the students the main ideas behind validation techniques, data preprocessing, dimension reduction, and many diverse models. These were the subjects covered the last time *Introduction to Big Data Modeling* was offered:

- Data preparation
- *k*-means
- Hierarchical clustering
- Principal components analysis
- Data acquisition
- Cross-validation
- *K* nearest neighbor
- Support vector machine
- Classification and regression trees
- Random forests
- Model ensemble

We spend roughly one week on each topic. Half of the time the students have expository lectures, where they learn the basics behind the methods they will learn, and half of the time they practice what they learn, applying their knowledge to solve real-world problems.

### 25.2.4 Ability to Communicate

The course also has an important communication component. In addition to the report the students need to write, their final project must be presented in front of an audience. Each group has 15–20 minutes to show their results to the other students in a slide presentation. Usually, the presentation is based on slide presentations. However, some students like to experiment and build dashboards to present their results.

The presentations are not graded only for their content, but also for their delivery. The students must demonstrate that they know the subject they are talking about. It is important for them to explain the dataset(s) they are using, how the data were obtained, what modeling worked best, what has failed, and so on.

In addition, the students are encouraged to ask questions of their peers, creating a respectful environment where everybody learns from each other.

## 25.3 Case Study

The case study we present here is the project the students have to complete on the web-scraping module. This is the fifth module of the course. By this point, students already know how to prepare the data for analysis (cleaning, transforming, and removing outliers) and have experience with three multivariate analysis methods: principal component analysis, *k*-means, and hierarchical clustering.

We believe the best way to engross the students in statistics is to engage them with projects to which they can relate. The project we propose for web scraping is to automatically extract data from Brazilian cities using Wikipedia in Portuguese. While the data are in Portuguese, it is very easy to translate those into English and understand what is happening in this project.

For this course, our programming language of choice is R. According to our knowledge, R is one the most used (if not the most used) languages in statistics departments around the world. Besides that, R is free, so money for software is not a bottleneck for the students. Other languages, such as python and Julia, could be used as well.

First, the students need to load the packages needed for the analysis. For this particular project, there are five packages needed:

- rvest: downloads web pages and extract the information we are interested in;
- dplyr: data manipulation, such as filtering and merging;
- ggplot2: used for plots;
- stringr: string cleaning;
- scales: formats plot labels.

Each package can be installed using the command install.packages(package), where package stands for each one of the five packages listed above. After the installation, they need to be loaded into R memory:

```
library(rvest)
library(dplyr)
library(ggplot2)
```

```
theme_set(theme_bw()) # set plot style
library(stringr)
library(scales)
```

The first step in the analysis is to define the web page address we want to download. We are interested in the link

```
https://pt.wikipedia.org/wiki/Lista_de_municípios_do_Brasil_por_
população, which has a table with Brazilian population data per city. In
the code below we inform it to R and download its contents using the
function read_html.
url <- "https://pt.wikipedia.org/wiki/
Lista_de_munic%C3%ADpios_do_Brasil_por_popula%C3
%A7%C3%A3o"
population <- url %>%
  read_html()
```

After downloading the web page, we need to extract the information we want from it. In this case, we will use the function html_table to obtain the table already present.

```
population <- population %>%
 html_table(fill=TRUE)

population <- population[[1]]
```

Since its column names are in Portuguese, we will translate them and check the result:

```
names(population) <- c("Position", "IBGE.Code", "City", "State",
"Population")

head(population)
```

| ## | Position | IBGE.Code | City | State | Population |
|---|---|---|---|---|---|
| ## 1 | 1º | 3550308 | São Paulo | São Paulo | 12252023 |
| ## 2 | 2º | 3304557 | Rio de Janeiro | Rio de Janeiro | 6718903 |
| ## 3 | 3º | 5300108 | Brasília | Distrito Federal | 3015268 |
| ## 4 | 4º | 2927408 | Salvador | Bahia | 2872347 |
| ## 5 | 5º | 2304400 | Fortaleza | Ceará | 2669342 |
| ## 6 | 6º | 3106200 | Belo Horizonte | Minas Gerais | 2512070 |

After the population data are obtained, we proceed in a similar fashion to obtain the city areas in square kilometers:

```
url <- "https://pt.wikipedia.org/wiki/Lista_de_munic%C3%ADpios_
brasileiros_por_%C3%A1rea_decrescente"

area <- url %>%
  read_html()

area <- area %>%
  html_table(fill=TRUE)

area <- area[[1]]
```

```
names(area) <- c("Position", "City", "IBGE.Code", "State", "Area")

head(area)
```

| ## | Position | City | IBGE.Code | State | Area |
|---|---|---|---|---|---|
| ## 1 | 1 | Altamira | 1500602 | Pará | 159 695,938 |
| ## 2 | 2 | Barcelos | 1300409 | Amazonas | 122 475,728 |
| ## 3 | 3 | São Gabriel da Cachoeira | 1303809 | Amazonas | 109 184,896 |
| ## 4 | 4 | Oriximiná | 1505304 | Pará | 170 602,992 |
| ## 5 | 5 | Tapauá | 1304104 | Amazonas | 89 324,259 |
| ## 6 | 6 | São Félix do Xingu | 1507300 | Pará | 84 212,426 |

Notice that both datasets have a column called IBGE.Code. IBGE stands for Instituto Brasileiro de Geografia e Estatística (English: Brazilian Institute of Geography and Statistics). Every city in Brazil has a unique IBGE code associated with it. This column is important to handle city misnaming when we join the datasets. The following code makes a new dataset with all the data we have so far, organizing the information according to the IBGE code:

```
brazil <- left join(population, area, by = "IBGE.Code")

head(brazil)
```

| ## | Position.x | IBGE.Code | City.x | State.x | Population | Position.y |
|---|---|---|---|---|---|---|
| ## 1 | 1 | 3550308 | São Paulo | São Paulo | 12252023 | 966 |
| ## 2 | 2 | 3304557 | Rio de Janeiro | Rio de Janeiro | 6718903 | 1236 |
| ## 3 | 3 | 5300108 | Brasília | Distrito Federal | 3015268 | 259 |
| ## 4 | 4 | 2927408 | Salvador | Bahia | 2872347 | 1903 |
| ## 5 | 5 | 2304400 | Fortaleza | Ceará | 2669342 | 3336 |
| ## 6 | 6 | 3106200 | Belo Horizonte | Minas Gerais | 2512070 | 3215 |

| ## | City.y | State.y | Area |
|---|---|---|---|
| ## 1 | São Paulo | São Paulo | 1 522, 986 |
| ## 2 | Rio de Janeiro | Rio de Janeiro | 1 182,296 |
| ## 3 | Brasília | Distrito Federal | 5 801,937 |
| ## 4 | Salvador | Bahia | 706,799 |
| ## 5 | Fortaleza | Ceara | 313,140 |
| ## 6 | Belo Horizonte | Minas Gerais | 330,954 |

Notice that some columns have duplicate data, such as City.x and City.y. It is redundant to have both in the same dataset. We remove the columns we do not want in the final dataset and rename them accordingly:

```
brazil <- brazil %>%
  select(City.x, State.x, Area, Population)

names(brazil) <- c("City", "State", "Area", "Population")

head(brazil)
```

| ## | City | State | Area | Population |
|---|---|---|---|---|
| ## 1 | São Paulo | São Paulo | 1 522,986 | 12252023 |
| ## 2 | Rio de Janeiro | Rio de Janeiro | 1 182,296 | 6718903 |
| ## 3 | Brasília | Distrito Federal | 5 801,937 | 3015268 |
| ## 4 | Salvador | Bahia | 706,799 | 2872347 |
| ## 5 | Fortaleza | Ceará | 313,140 | 2669342 |
| ## 6 | Belo Horizonte | Minas Gerais | 330,954 | 2512070 |

However, area is not ready for analysis, as there are spaces and decimal separators in it. This will make R understand area not as numbers, but as strings. Therefore, we need to clean these values and transform them into a numeric variable.

```
brazil <- brazil %>%
  # Area transformation
  mutate(Area = str_replace(Area, "[[:space:]]", "")) %>%
  mutate(Area = str_replace(Area, ", ", ".")) %>%
  mutate(Area = as.numeric(Area))

head(brazil)
```

| ## | City | State | Area | Population |
|---|---|---|---|---|
| ## 1 | São Paulo | São Paulo | 1 522,986 | 12252023 |
| ## 2 | Rio de Janeiro | Rio de Janeiro | 1 182,296 | 6718903 |
| ## 3 | Brasília | Distrito Federal | 5 801,937 | 3015268 |
| ## 4 | Salvador | Bahia | 706,799 | 2872347 |
| ## 5 | Fortaleza | Ceará | 313,140 | 2669342 |
| ## 6 | Belo Horizonte | Minas Gerais | 330,954 | 2512070 |

Now the dataset is ready for analysis. One of the tasks proposed to the students is to check if there is a linear relationship between city area and population. After all, it is expected that the larger the area of the city, the more people will live there. However, the plot does not reflect this (Figure 25.2):

```
ggplot(brazil, aes(x=Area, y=Population)) +
  geom_point() +
  labs(x="Area (km^2)", y="Population")
```

Most students notice that the reason behind this behavior is the population density. Hence, they are asked to look for the cities' densities and check if they are unequal across the dataset:

```
brazil <- brazil %>%
  mutate(Density = Population/Area)

brazil %>%
  arrange(desc(Density)) %>%
  head(5)
```

| ## | City | State | Area | Population | Density |
|---|---|---|---|---|---|
| ## 1 | Taboão da Serra | São Paulo | 20.478 | 289664 | 14145.13 |
| ## 2 | Diadema | São Paulo | 30.650 | 423884 | 13829.82 |
| ## 3 | São João de Meriti | Rio de Janeiro | 34.838 | 472406 | 13560.08 |
| ## 4 | Carapicuíba | São Paulo | 34.967 | 400927 | 11465.87 |
| ## 5 | Osasco | São Paulo | 64.935 | 698418 | 10755.65 |

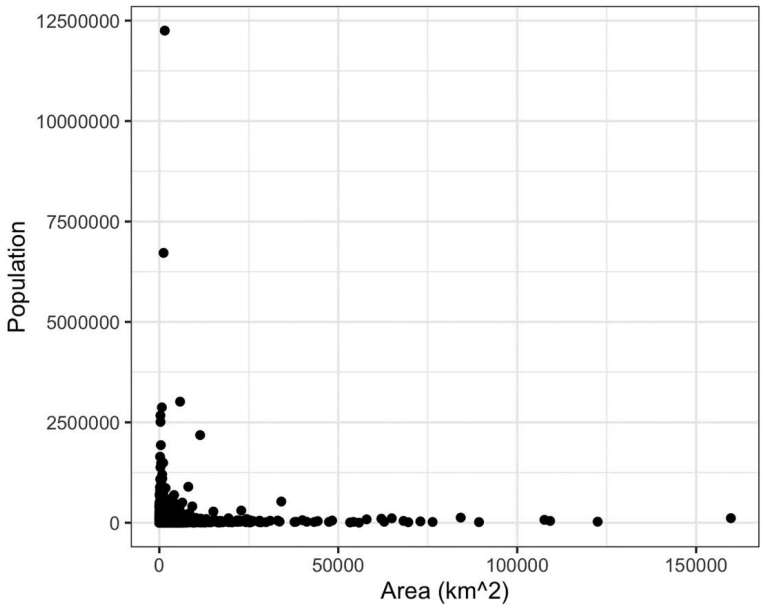


**FIGURE 25.2**
Relationship between population and area for the Brazilian cities.

```
brazil %>%
  arrange(desc(Density)) %>%
  tail(5)
```

| ## | City | State | Area | Population | Density |
|---|---|---|---|---|---|
| ## 5566 | Ipiranga do Norte | Mato Grosso | NA | 7667 | NA |
| ## 5567 | Itanhangá | Mato Grosso | NA | 6737 | NA |
| ## 5568 | Paraíso das Águas | Mato Grosso do Sul | NA | 5555 | NA |
| ## 5569 | Pinto Bandeira | Rio Grande do Sul | NA | 3003 | NA |
| ## 5570 | Aroeiras do Itaim | Piauí | NA | 2551 | NA |

Many other tasks are presented to the students regarding to this specific dataset. These problems are made to challenge them, in order to make them relate different subjects and apply different statistical methods.

Note that in this chapter, we only deal with obtaining data through web scraping. As the course progresses, new topics are introduced to students. As they are cumulative issues, each new problem involves understanding what is being addressed and how it can be related to the issues addressed up to that point.

## 25.4 Final Remarks

We presented some considerations for our course *Introduction to Big Data Modeling*. As we stated previously, this course follows the ASA Guidelines for undergraduate programs in statistics.

Student evaluations indicate students are satisfied with this course contents. "The content of this course is very interesting and important for students who have worked with data science," "Excellent course! It covers a lot of knowledge in R programming," and "New discipline, but recommended for everyone, since it is a current and well-spoken subject worldwide" are some of the testimonials we have received so far.

We believe *Introduction to Big Data Modeling* is well implemented at our university. It was first offered in 2015 for students in the Statistics Department; 2019 was the first year the course was offered for the students enrolled in the Actuarial Science Department, and this met with huge success as well.

Our future plans for this course include expanding it from a one-semester to a two-semester course. Its first part will remain the same, but we plan to add more advanced topics in its second part. For example, time series are not covered in the present course. More recent methods, such as deep learning, are not considered either. Therefore, an expansion could be a good idea, since it would expose the students to more techniques and enable them to analyze a wider range of data.

This course is on par with the most recent introductory machine learning courses around the world. Since all data and software we use are free, budget is not a concern. However, some investment of time is necessary, since most professors are not used to these new learning approaches.

## References

Curriculum Guidelines for Undergraduate Programs in Statistical Science. http://www.amstat.org/education/pdfs/guidelines2014-11-15.pdf.

Donoho, D. 2017. "50 years of data science." *Journal of Computational and Graphical Statistics* 26 (4): 745–766.

Grolemund, G., and H. Wickham. 2011. "Dates and times made easy with lubridate." *Journal of Statistical Software* 40 (3): 1–25. http://www.jstatsoft.org/v40/i03/.

Healy, K. J. 2018. *Data Visualization: A Practical Introduction*. Princeton University Press. https://books.google.com.br/books?id=o6BYtgEACAAJ.

Hicks, S. C. and R. A. Irizarry. 2016. "A guide to teaching data science." *The American Statistician* 72 (4): 382–391.

Lazar, N. A., J. Reeves, and C. Franklin. 2011. "A capstone course for undergraduate statistics majors." *The American Statistician* 65 (3): 183–189. https://doi.org/10.1198/tast.2011.10240.

Massicotte, P., and D. Eddelbuettel. 2020. *GtrendsR: Perform and Display Google Trends Queries*. https://CRAN.R-project.org/package=gtrendsR.

Wagaman, A. 2016. "Meeting student needs for multivariate data analysis: A case study in teaching an undergraduate multivariate data analysis course." *The American Statistician* 70 (4): 405–412. https://doi.org/10.1080/00031305.2016.1201005.

Wickham, H. 2016. *Ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag, New York. https://ggplot2.tidyverse.org.

Wickham, H. 2019a. *Rvest: Easily Harvest (Scrape) Web Pages*. https://CRAN.R-project.org/package=rvest.
Wickham, H. 2019b. *Stringr: Simple, Consistent Wrappers for Common String Operations*. https://CRAN.R-project.org/package=stringr.
Wickham, H., R. François, L. Henry, and K. Müller. 2020. *Dplyr: A Grammar of Data Manipulation*. https://CRAN.R-project.org/package=dplyr.
Wickham, H., and D. Seidel. 2020. *Scales: Scale Functions for Visualization*. https://CRAN.R-project.org/package=scales.
Wickham, H., and G. Grolemund. 2017. *R for Data Science*. O'Reilly Media. https://books.google.com.br/books?id=-7RhvgAACAAJ.