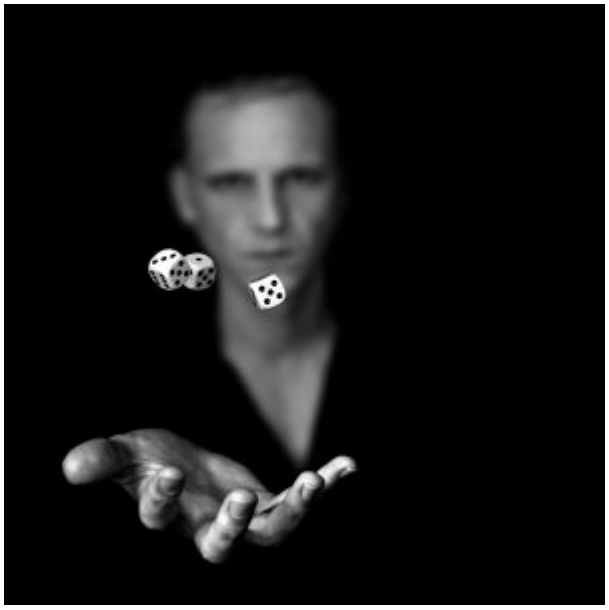
Answer probability questions with simulation (part-2)



This is the second exercise set on answering probability questions with simulation. Finishing the first exercise set is not a prerequisite. The difficulty level is about the same – thus if you are looking for a challenge aim at writing up faster more elegant algorithms.

As always, it pays off to read the instructions carefully and think about what the solution should be before starting to code. Often this helps you weed out irrelevant information that can otherwise make your algorithm unnecessarily complicated and slow.

Answers are available here.

Exercise 1
If you take cards numbered from 1-10 and shuffle them, and lay them down in order, what is the probability that at least one card matches its position. For example card 3 comes down third?

Exercise 2
Consider an election 3 candidates and 35 voters and who casts his ballot randomly for one of the candidates. What is the probability of a tie for the first position?

Exercise 3
If you were to randomly find a playing card on the floor every day, how many days would it take on average to find a full standard deck?

Exercise 4
Throw two dice. What is the probability the difference between them is 3, 4, or 5 instead of 0, 1, or 2?

Exercise 5
What is the expected number of distinct birthdays in a group of 400 people? Assume 365 days and that all are equally likely.

Exercise 6
What is the probability that a five-card hand in standard deck of cards has exactly three aces?

Learn more about probability functions in the online course Statistics with R – Advanced Level. In this course you will learn how to

- work with different binomial and logistic regression techniques,
- know how to compare regression models and choose the right fit,
- and much more.

Exercise 7
Randomly select three distinct integers a, b, c from the set {1, 2, 3, 4, 5, 6, 7}.
What is the probability that a + b > c?

Exercise 8
Given that a throw of three dice show three different faces what is the probability if the sum of all the dice is 8.

Exercise 9
Throwing a standard die until you get 6. What is the expected number of throws (including the throw giving 6) conditioned on the event that all throws gave even numbers.

Exercise 10
Choose two-digit integer at random, what is the probability that it is divisible by each of its digits. This is not exactly a simulation proplem – but the concept is similar make R do the hard work.

(Picture by Gil)

R with remote databases Exercises (Part-2)

This is common case when working with data that your source is a remote database. Usual ways to cope this when using R is either to load all the data into R or to perform the heaviest joins and aggregations with SQL before loading the data. Both of them have cons: the former one is limited by the memory

capacity and may be very slow and the later forces you to use two technologies thus is more complicated and prone to errors. Solution to these problems is to use dplyr with dbplyr to communicate with database backend. This allows user to write dplyr code that is translated to SQL and executed at database server. One can say that this combines advantages of the two standard solutions and gets rid of their disadvantages.

This is the second part of R with remote databases series. For other parts follow the tag databases.

The reader is assumed to know basics of dplyr and SQL. If you want to practice dplyr first there is great series of exercises Data wrangling: Transforming available. For quick introduction to dplyr with database backend I recommend this vignette.

Answers to the exercises are available here.

If you obtained a different (correct) answer than those listed on the solutions page, please feel free to post your answer as a comment on that page.

Exercise 1
Load libraries: dplyr, dbplyr, DBI, RSQLite, nycflights13. Create a connection to temporal in-memory SQLite database and load to it two copies of data set nycflights13::flights – one with no indexes and another with index at carrier.

Exercise 2
Compare execution time for counting a number of flights per carrier for indexed and not indexed table. (HINT: You can use st <- Sys.time(); **code**; Sys.time - st or microbenchmark package.)

Exercise 3
Compare query plans for queries from Exercise 2.

Exercise 4

Check how dbplyr translates as.character to SQL.

Exercise 5
Check how dbplyr translates substr to SQL depending on values of parameters. What is the relation between R and SQL version?

Exercise 6
Check how dbplyr translates ^ to SQL. Check how does it work with SQLite.

Exercise 7
Check how dbplyr translates mean to SQL. What happens when you specify na.rm or trim parameter?

Learn more about Data Pre-Processing in the online course R Data Pre-Processing & Data Management – Shape your Data!. In this course you will learn how to:

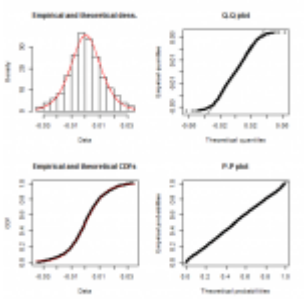
- import data into R in several ways while also beeing able to identify a suitable import tool
- use SQL code within R
- And much more

Exercise 8
Calculate mean value of dep_delay per carrier. Compare results for the table in the database and for local one.

Exercise 9
In database calculate number of destination airports with airport code starting with 'A'. Use wildcard matching.

Exercise 10
In database concatenate origin and dest to one column separated by '-'.

Generalized linear functions (Beginners)



On this set of exercises, we are going to use the lm and glm functions to perform several generalized linear models on one dataset.

Since this is a basic set of exercises we will take a closer look at the arguments of these functions and how to take advantage of the output of each function so we can find a model that fits our data.

Before starting this set of exercises, I strongly suggest you look at the R Documentation of lm and glm.

Note: This set of exercises assume that you have a basic understanding of generalized linear models.

Answers to the exercises are available here.

If you obtained a different (correct) answer than those listed on the solutions page, please feel free to post your answer as a comment on that page.

The dataset we will be using contains information from passengers of the Titanic including if they survived or not.

To obtain the data run these lines of code.

```
if     (!'titanic'     %in%     installed.packages())
install.packages('titanic')
library(titanic)
DATA <- titanic_train[,-c(1,4,9,11)]
```

Exercise 1

Linear regression

1. Use DATA to create a linear model using the function lm with the variables Age and Fare as independent variables and Survived as the independent one. Save the regression in an object called lm_reg

2. Use the function glm to perform the same task and save the regression in an object called glm_reg

Exercise 2

If you print any of the previous objects you will realize that there's not much information about the performance of the models, fortunately summary is a great function to find out more about any statistical model you preform to a dataset. Depending on the model summary will produce different outputs.

- Apply summary to lm_reg and to glm_reg. You will find a slight difference between both of the outputs, that is because glm is more flexible than lm.

Exercise 3

So far we have been assuming (incorrectly) that the dependent variable (Survived) follows a normal distribution and that's why we have been performing a linear regression. Obviously Survived follows a binomial distribution, there are only two options either the passenger survived (1) or the passenger wasn't that lucky and he died (0). Since the data has a binomial distribution we should perform a logistic regression, to do this use the function glm to perform a logistic regression using Age and Fare as independent variables and

save it in an object called bin_model. Hint: Define the value of the argument family properly.

Exercise 4

Inside the family attribute you can always specify a particular link, in case you don't a default link will be associated depending on the family you chose.

1. To find out the default link associated to a certain family, you can write the family name followed by a parenthesis (Ex. gaussian(). Find the default link associated to the binomial family.
2. Create a probit model with the same variables used in bin_model and save it in an object called bin_probit_model.

Exercise 5

Findind the right model requires to compare different models and choose the best, although there are many performance measures, for now we will use the AIC as our measure (smaller AIC are better). This means that bin_model is better than bin_probit_model, so let's keep working with bin_model.

Until now intercept variable has been part of the models. Create a logistic regression with the same variables but with no intercept.

Exercise 6

Impute data. If you run the summary function to any of the previous models you will find out that 177 observations have been deleted due to missingness. This happens because the glm function has as default argument na.acton ="na.omit". This make easier to run a model with messier data, but that is not always great. You want to have full control an understanding of what does the function is doing.

1. There are some missing values in age, replace this values with the median.

2. Update the glm_model with the updated data, specify na.action='na.fail' This will assure us that the dataset has no missing values, otherwise it will show an error.

Learn more about evaluating different statistical models in the online courses Linear regression in R for Data Scientists and Structural equation modeling (SEM) with lavaan. These courses cover different statistical models that can help you choose the right design for your solution.

Exercise 7

Add polynomial independent variables. Some variables have a quadratic interaction between them and the dependent variable, this can be solved by specifying in the formula of the model a quadratic interaction.

Add a quadratic interaction for the variable Fare into the current model, specified in glm_model

Exercise 8

Add categorical variables. Add Sex as an independent variable into the current model specified in glm_model. Note that Sex is not a numeric variable.

Exercise 9

Now that we have found a good model that fits our data, so it's time to use the predict function to find how good the model predicts in our own data. Use the function predict to find the prediction of the model in DATA and save it in Pred.default

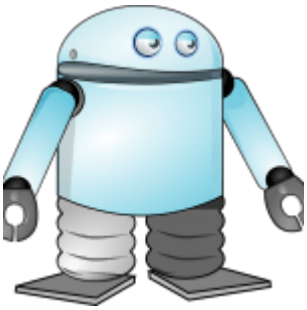
Exercise 10

Pred.default shows the predicted values under the link transformation, in this case logit. This is not easily interpretable, to fix this problem we can specify the type of prediction we want.

Obtain the predictions as probability values.

Exta: What's the percentage accuracy of this model if we assigned as died (0) if the predicted probability is less than 0.5 and survived (1) otherwise?

Applying machine learning algorithms – exercises



INTRODUCTION

Dear reader,

If you are a newbie in the world of machine learning, then this tutorial is exactly what you need in order to introduce yourself to this exciting new part of the data science world.

This post includes a full machine learning project that will guide you step by step to create a "template," which you can use later on other datasets.

Before proceeding, please follow our short tutorial.

Look at the examples given and try to understand the logic behind them. Then try to solve the exercises below using R and without looking at the answers. Then see the solutions to check your answers.

Exercise 1

Create a list named “control” that runs a 10-fold cross-validation. **HINT**: Use trainControl().

Exercise 2

Use the metric of “Accuracy” to evaluate models.

Exercise 3

Build the “LDA”, “CART”, “kNN”, “SVM” and “RF” models.

Exercise 4

Create a list of the 5 models you just built and name it “results”. **HINT**: Use resamples().

Learn more about machine learning in the online course Beginner to Advanced Guide on Machine Learning with R Tool. In this course you will learn how to:

- Create a machine learning algorithm from a beginner point of view
- Quickly dive into more advanced methods in an accessible pace and with more explanations
- And much more

This course shows a complete workflow start to finish. It is a great introduction and fallback when you have some experience.

Exercise 5

Report the accuracy of each model by using the summary function on the list “results”. **HINT**: Use summary().

Exercise 6

Create a plot of the model evaluation results and compare the spread and the mean accuracy of each model. **HINT**: Use dotplot().

Exercise 7

Which model seems to be the most accurate?

Exercise 8

Summarize the results of the best model and print them. **HINT**: Use print().

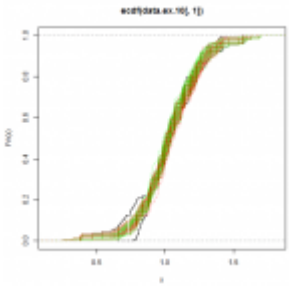
Exercise 9

Run the "LDA" model directly on the validation set to create a factor named "predictions". **HINT**: Use predict().

Exercise 10

Summarize the results in a confusion matrix. **HINT**: Use confusionMatrix().

Probability functions advanced



In this set of exercises, we are going to explore some applications of probability functions and how to plot some density functions. The package MASS will be used in this set.

Note: We are going to use random numbers functions and random processes functions in R such as runif. A problem with these

functions is that every time you run them you will obtain a different value. To make your results reproducible, you can specify the value of the seed using set.seed('any number') before calling a random function. (If you are not familiar with seeds, think of them as the tracking number of your random number process.) For this set of exercises, we will use set.seed(1). Don't forget to specify it before every exercise that includes random numbers.

Answers to the exercises are available here. If you obtained a different (correct) answer than those listed on the solutions page, please feel free to post your answer as a comment on that page.

Exercise 1

Generating dice rolls Using the functions runif and round, simulate the results of 100 dice rolls.

Exercise 2

Let's assume that we want to simulate a game in which we throw an unfair coin (success probability is 0.48) 10 times and you win $10 every time the result is tails and lose $10 when the result is heads. Simulate this game1000 time using rbinom, and find the expected amount of money you will gain or lose in this game using the simulated values.

Exercise 3

Simulate an experiment of throwing one dice 30 times using the function rmultinom, and find out how many 6's are in the simulated sample.

Exercise 4

Obtain a vector that shows how many 1's, 2's,….6's were obtained in the previous simulation.

Exercise 5

Simulate normal distribution values. Imagine a population in which the average height is 1.70 m with a standard deviation of 0.1. Use rnorm to simulate the height of 1000 people and save it in an object called heights.

a) Plot the density of the simulated values.
b) Generate 10000 values with the same parameters and plot the respective density function on top of the previous plot in red to differentiate it.

This plot will show you how much a sample with 10000 simulations approximate to the real normal distribution.

Exercise 6

Find the 90% interval of a population with mean = 1.70 and standard deviation = .1

Exercise 7

Simulate 100000 people with height (cm) and weight (kg) using the function mvrnorm with mu = c(170, 60) and
Sigma = matrix(c(10,17,17,100), nrow = 2), and save it in an object called population.

Apply the function summary to population to get an idea of the values created.

Learn more about probability functions in the online course Statistics with R – Advanced Level. In this course you will learn how to

- work with different binomial and logistic regression techniques,
- know how to compare regression models and choose the right fit,
- and much more.

Exercise 8

Plotting bivariate distribution. Use the function kde2d to generate a two-dimensional kernel density of the matrix population and plot the values using persp.

Exercise 9

Simulating with a Bayesian approach. Unlike the frequentist statistics approach, Bayesian statistics assume the parameters of a distribution are a random variable with its own distribution. Let's simulate a poisson variable.

a) Simulate a gamma variable with shape = 20 and scale = 0.5
b) Simulate using the previous value a poisson random variable

Exercise 10

Simulating one variable doesn't make sense if you want to know the properties of a certain distribution. Repeat the previous simulation but create 100 poisson variables and plot the distribution.

Data wrangling : Cleansing – Regular expressions (3/3)

Data wrangling is the process of importing, cleaning, and transforming raw data into actionable information for analysis. It is a time-consuming process that is estimated to take about 60-80% of analysts' time. In this series, we will go through this process. It will

be a brief series with the goal of crafting the reader's skills in data wrangling. This is the fourth part of the series and it aims to cover the cleaning of the data used. In previous parts, we learned how to import, reshape, and transform data. The rest of the series will be dedicated to the data cleansing process. In this post, we will go through the regular expressions, which is a sequence of characters that define a search pattern, mainly
for use in pattern matching with text strings. In particular, we will cover the foundations of regular expression functions.

Before proceeding, it might be helpful to look over the help pages for the grep, sub, gsub, strsplit, .

Moreover, please load the following library.

```
install.packages("stringr")
library(stringr)
```

Answers to the exercises are available here. If you obtained a different (correct) answer than those listed on the solutions page, please feel free to post your answer as a comment on that page.

Exercise 1

Find the cars that are Mercedes-Benz (match the pattern 'Merc').
Hint: The names of the cars can be retrieved from the command rownames(mtcars)

Exercise 2

Find the cars that are **not** Mercedes-Benz.

Exercise 3

Find the cars that are Mercedes-Benz (match the pattern 'Merc'), but with a logical output.

Exercise 4

Find the number of Mercedes-Benz in the data set.

Exercise 5

Replace the first ‘a’ of every car with an ‘e’.

Learn more about Text analysis in the online course Text Analytics/Text Mining Using R. In this course you will learn how create, analyse and finally visualize your text based data source. Having all the steps easily outlined will be a great reference source for future work.

Exercise 6

Replace all ‘a’s of every car with ‘

Exercise 7

Separate the brand from the model. (e.g. “Mazda RX4” -> “Mazda” “RX4”).

Exercise 8

Find the cars that are Mercedes-Benz (use the str_detect function).

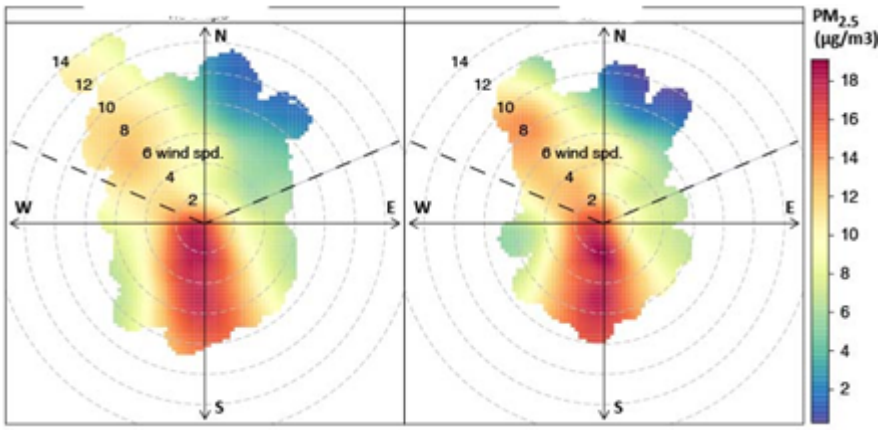
Exercise 9

Extract the ‘Merc’ string from the cars that contain it.

Exercise 10

Replace the ‘Merc’ string from the cars that contain it with ‘Mercedes-Benz’.

Working with air quality and meteorological data Exercises (Part-3)



Atmospheric air pollution is one of the most important environmental concerns in many countries around the world, and it is strongly affected by meteorological conditions. Accordingly, in this set of exercises we use openair package to work and analyze air quality and meteorological data. This packages provides tools to directly import data from air quality measurement network across UK, as well as tools to analyse and producing reports.

In the previous exercises set we used data from MY1 station to see how to import data and extract basic statistical information from data. Then we practiced a few basic functions that are available in openair package. In this exercise set we will practice more advance functions that are very useful for air quality data analysis.

Answers to the exercises are available here.

For other parts of this exercise set follow the tag openair

Please load the package openair before starting the exercises.

Exercise 1
In the first exercise we will use timeVariation function to see the variation of a pollutant by time of day and day of week which can reveal useful information. For example, o3 is

produced only during daytime so it is typically expected to be maximum in the afternoon and reaches its minimum during nighttime. The timeVariation function produces four plots: day of the week variation, mean hour of day variation and a combined hour of day – day of week plot and a monthly plot.

Use timeVariation function to plot timeseries for pm10 and o3 for my1data that was used in the previous exercise sets.

Exercise 2
timeVariation function is also flexible to subset data based on the given conditions. For example one can set the conditions to consider only those hours when wind speed and direction is in a specific range.

Use timeVariation function to plot timeseries for pm10 and o3 for my1data. Only consider those hours when ws > 5 and wd is between 50 and 200.

Exercise 3
It is also possible to see the time series of multiple variables using timeVariation. For example the production of o3 is strongly depends on its precursors including, nox, no2, and co. So it would be very useful to see the variation of o3 concentration with its precursors.

Use timeVariation function to plot time series for o3, nox, no2, and co for my1data.

You can use Air Quality Data and weather patterns in combination with spatial data visualization, **Learn more** about spatial data in the online course [[Intermediate] Spatial Data Analysis with R, QGIS & More](). this course you will learn how to:

- Work with Spatial data and maps
- Learn about different tools to develop spatial data next

to R
- And much more

Exercise 4
Another interesting function that is available in openair is scatterPlot function. The purpose of this function is to make it straightforward to consider how variables are related to one another in a way consistent with other openair functions.

use scatterPlot to show scatter plot of daily o3 concentration for the month of April with different levels of ambient temperature.

Exercise 5
use scatterPlot to show scatter plot of nox vs no2 concentration by the level of temperature in each season.

Big Data analytics with RevoScaleR Exercises-2

In the last set of exercises , you have seen the basic functionalities of RevoScaleR .In this exercise set we will explore RevoScaleR further.
get the Credit card fraud data set from revolutionanalytics and lets get started
Answers to the exercises are available here.Please check the

documentation before starting these exercise set

If you obtained a different (correct) answer than those listed on the solutions page, please feel free to post your answer as a comment on that page.

Exercise 1
RevoScaleR provides option to convert a dataframe into a xdf file,which you might need while storing temporary data frame that you create during analysis work .
Now Create an XDF file from airquality dataset

Exercise 2
In the previous set of exercise you have seen rxHistogram briefly,Now we will see how to get meaningful information from large dataset with a visualization .
create a scatterplot from the xdf file for Balance vs numTrans where numTrans is greater than 50 and have more than 50 creditline and the fraudrisk is 1 .

Exercise 3
Good thing about RevoScaleR is that it comes with a sample data directory,find the sample data directory by rxGetoptions and use claims.text and convert it into xdf file .You have already used rxImport ,which can be used but for plain text data there is an highly optimized method in RevoscaleR.Please use that

Exercise 4
if you check the sampledatadir you can see that there are 10 csv files like moredefaulsmall2000.csv etc . rxImport can create a single xdf by joining them row wise .
Please create a new xdf which will contain all the data from moredefaultsmall2000.csv to moredefaultsmall2010.csv.
You can write a loop or do it in a functional way , if you have followed my exercises on functional programming (1 & 2) , I hope you know how to achieve this without loop .

Exercise 5

We will see how rxDataStep can create new variables with complex calculations. on the creditcard fraud xdf create a column domTrans which is numTrans -numIntlTrans and create another
column balance per domTran which is Balance/domTrans. This exercise will show you how we can use temp variables to create more complex new columns from existing columns

Exercise 6
in rxDataStep an important feature is transform functions . One thing to remember that when you use a transform function ,the transform function sees a chunk of data at a time , so if you require any value for the whole dataset ,you need to create it before the transformfunction ,not within the transform function .
Now create the z score of balance in the credit card fraud xdf file,which you have already created in the first set of exercise.

The next 3 exercise will give you a basic idea on how to split xdf into training /test set , we will see a bit deeper into rxLinMod as well.Please remember that the goal
of this exercise is not to predict but to make you aware of few important details of modelling by revoscaleR

Exercise 7
split the credit card fraud xdf into two random parts with 25 percent being the test data and 75 percent is the training data.
Hint -You need to use splitByFactor and create it using transforms .Use a Seed to make this reproducible

Learn more about importing big data in the online course Data Mining with R: Go from Beginner to Advanced. In this course you will learn how to

- work with different data import techniques,

- know how to import data and transform it for a specific moddeling or analysis goal,
- and much more.

Exercise 8
In the last set you used rxLinMod ,use the same expression but use cube =true as a parameter . You may need to tweak around to make it work , as the first expression should be categorical when using cube . Can you see the difference between both models .

Exercise 9
How do you define a linear regression where you want to analyze whether fraudrisk depends only on the interaction of gender and balance and how do you define if you want to check fraudrisk's dependency on the interaction as well as their individual contribution .

Exercise 10
You might want to check fraudrisk for a different segment of balance . Create 5 different segments of 0-10k,10k-20k and so on and check the Linear Regression result .

R with remote databases Exercises (Part-1)

This is common case when working with data that your source is a remote database. Usual ways to cope this when using R is either to load all the data into R or to perform the heaviest joins and aggregations with SQL before loading the data. Both of them have cons: the former one is limited by the memory capacity and may be very slow and the later forces you to use two technologies thus is more complicated and prone to errors. Solution to these problems is to use dplyr with dbplyr to communicate with database backend. This allows user to write dplyr code that is translated to SQL and executed at database server. One can say that this combines advantages of the two standard solutions and gets rid of their disadvantages.

This is the first part of R with remote databases series. For other parts follow the tag databases.

The reader is assumed to know basics of dplyr and SQL. If you want to practice dplyr first there is great series of exercises Data wrangling: Transforming available. For quick introduction to dplyr with database backend I recommend this vignette.

Answers to the exercises are available here.

If you obtained a different (correct) answer than those listed on the solutions page, please feel free to post your answer as a comment on that page.

Exercise 1
Load libraries: dplyr, dbplyr, DBI, RSQLite, nycflights13. Create a connection to temporal in-memory SQLite database (the database will be created on-the-fly so you do not have to take care of this).

Exercise 2
Upload data sets nycflights13::flights and nycflights13::planes to the database as non temporal tables.

Exercise 3

List names of columns in flights table (hint: There is a function in DBI package for this).

Exercise 4
Use SQL query to count number of flights per carrier and pull it to a local tibble.

Exercise 5
Do the same thing using dplyr verbs instead of SQL.

Exercise 6
Calculate number of flights, mean and total distance per plane, discard records with NA at tailnum column and save it to temporal table.

Learn more about Data Pre-Processing in the online course R Data Pre-Processing & Data Management – Shape your Data!. In this course you will learn how to:

- import data into R in several ways while also beeing able to identify a suitable import tool
- use SQL code within R
- And much more

Exercise 7
List all the tables in the database (hint: There is a function in DBI package for this).

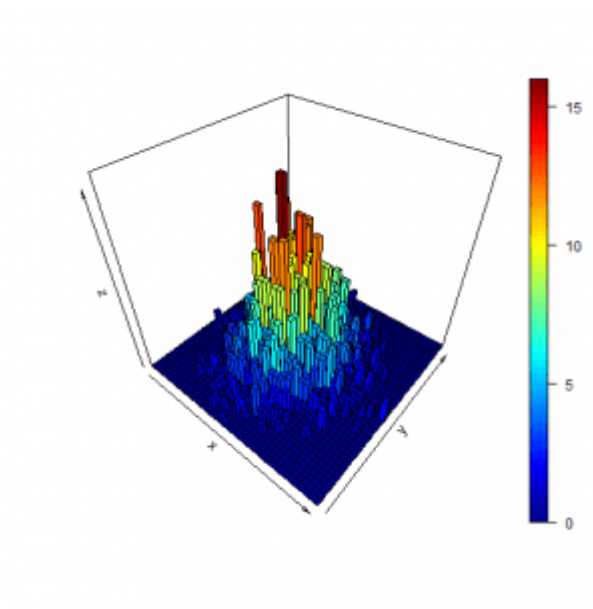
Exercise 8
Use head(), tail() and nrow() to investigate table planes. Do you understand why the latter two do not work?

Exercise 9
Join the table from exercise 6 with planes table and without pulling the data to local frame find manufacturers and models of 10 planes with the highest total flown distance.

Exercise 10

Check what is the actual SQL query generated by the code you have created in exercise 9.

Hacking statistics or: How I Learned to Stop Worrying About Calculus and Love Stats Exercises (Part-8)



Statistics are often taught in school by and for people who like Mathematics. As a consequence, in those class emphasis is put on leaning equations, solving calculus problems and creating mathematics models instead of building an intuition for probabilistic problems. But, if you read this, you know a bit of R programming and have access to a computer that is really good at computing stuff! So let's learn how we can tackle useful statistic problems by writing simple R query and how to think in probabilistic terms.

In today's set we take a break of hypothesis testing and we come back to the fundamental of statistics: the probability. Precisely, in this set, you will see how to compute probability of complex events, use conditional and marginal distribution function and learn to sample from and plot a multivariate distribution function.

Answers to the exercises are available here.

For other parts of this exercise set follow the tag Hacking stats

Exercise 1
So far we know that probabilities take a value between 0 and 1. We know that probabilities of realization of single events that form a set can be added together to compute the probability of realization of any event in that set. For example, the probability of getting hit by a bus on a given day or bitten by a shark is equal to the sum of those probabilities. However, we can say that because it's almost impossible that both event happen on the same day (if you know somebody who got bitten by a shark, survived, then got hit by a bus, please stay far from them for your own safety!). Those kinds of event are called mutually exclusive and can be identified by looking at the Venn diagram of the outcome. More info here. For those interested, when two evens are not mutually exclusive, we can still add their probabilities together to get the probability of realization of one event or the other, but we must subtract the probability of getting both even to the total. The next exercise should give you an idea why we must subtract this value from the total.

The quality assurance department of a video game studio classify found bug in two categories: graphic issues or collision bug. One of the tester created this dataset compiling the bugs he found during an average workday.

1. Use the VennDiagram package to draw the Venn Diagram of the dataset.
2. What is the probability of finding a graphic issues uniquely? Of getting only a collision bug?
3. What is the probability that the tester find a graphic bug that is also a collision bug?
4. What is the probability that the tester find a graphic bug or a collision bug?

Exercise 2

If we have two events A and B, we know how to compute the probability that A or B happen. Now if you want to know the probability of observing A and B, there's two possible scenarios: the one where the realization of A influence the probability of realization of the event B and the one where the probability of B stay the same whether A happen or not. This last case is the easier to compute: we just have to multiply both probabilities to get the probability of realization of both events.

This result can be extended to more than two event. For example, if you flip a coin three times and want to know the probability to get three heads, you know that each coin flip result doesn't influence the next result. As consequence you can just multiply the probability of each event, in this case 0.5*0.5*0.5=0.125 to know the probability of this particular result.

1. Sample with replacement 500 integers between 1 and 10 and store the result in a vector called Event.A.
2. Sample with replacement 500 integers between 1 and 5 and store the result in a vector called Event.B.
3. If each element in both vector represent the result of a simultaneous draw. Empirically, what is the probability to draw the number 5 in both vectors, at the same time? What is the probability to draw a 1 in the first vector and a number bigger than 3 in the second?
4. Use the multiplication rule to compute the probability of those events and compare the results of the last exercise.

Exercise 3

When the realization of an event A change the probability of realization of the event B we estimate what is called a conditional probability. To do so, we use the same process than we used to estimate probability, but since the event A change the possible outcome of event B, we will used the

number of those possible outcomes as denominator in our formula. So the general formula for estimation of probability #of observation of B/total number of observations become #of observations of B when A happen/total number of observations when A happen. Here's some more formal definition here.

1. Load this dataset and explore it (make a histogram and list the unique observed value).
2. Compute the probability to observe each value.
3. There's seems to be two sub-processes that compose those random events. Let's assume that this dataset represent a lottery where you have 1 chance out of 100 to get a bonus that multiply by 10 your prize and that this bonus appear only in a winning situation. In this case, we could be interested to know the probability of winning and not having the bonus. Use the dataset to estimate the probability of those individual events.
4. Compute the probabilities of winning each amount when the bonus is applied.

Exercise 4

In a rural fair, people can pay 5 dollars to play a game where they choose to open one of three doors and pick a plastic ball from a closed box that sits behind the door. If the ball is red, they win 50 dollars and if the ball is blue, they win nothing. Each box contains 50 balls, but the amount of red ball change from one box to the other. A bored statistician have spent an afternoon compiling which door has been chosen by 450 players and if they won.

1. Load this dataset.
2. Estimate the probability of winning at this game.
3. Estimate the probability of winning at this game, if you choose the first door, the second door or the third door.
4. Create a contingency table of this situation.
5. Use the table to compute the conditional probability of winning if someone chose the first door, the second or

the third door.

Just as for the ordinary probability we can create a distribution from the conditional probabilities to better understand how a random process behave. The easiest way to compute such a distribution is to use a contingency table where all the outcome of two even are listed in the margin and the elements are the number of observations of each combination of outcome. The conditional distribution if an outcome Ai happened correspond to the ECDF computed by using the observation on the row or column of Ai.

Learn more about probability functions in the online course Statistics with R – Advanced Level. In this course you will learn how to:

- Work with about different binomial and logistic regression techniques
- Know how to compare regression models and choose the right fit
- And much more

Another useful distribution is the marginal distribution, which is the distribution of the individual event A and B. The name marginal come from the fact that when using a contingency table to estimate it, we must use the total of each rows and columns to compute the ECDF and those values are often put in the margins. The next exercise should help you get familiar with those concepts.

Exercise 5

A sample of 50 articles from three websites on the same subject has been analyzed by a professional facts checker to see the quality of their news coverage. The news has been classify in three categories: factually correct, mostly correct and fake news. The following dataset show the result of his work.

1. What is the probability of getting factually correct, mostly correct and fake news by looking at a random article from one of those sites?
2. What is the probability of reading a fake news from the first website?
3. What is the probability of reading the second website if you are reading a factually correct article?
4. What is the marginal distribution in this situation?
5. What is the conditional distribution for the mostly correct news?

Let’s look at the multivariate normal distribution and how the marginal and the conditional distribution are used in this case. Basically, a multivariate normal distribution is a function of dimension higher than 1 whose component are normaly distributed.

Exercise 6

1. Generate 2000 points from a standard normal distribution and store the results in a vector called x.
2. Generate 2000 points from a normal distribution of mean 10 and a standard deviation of 5 and store the result in a vector called y.
3. Create a matrix with two columns x and y which will be the coordinate of 2000 points.
4. Make a basic plot of the points in the last matrix and draw the histogram of both x and y matrix.

We know the marginal distributions of the multivariate normal distribution of the last exercise: they are the distribution of the x and y variables. Fun fact: the projection of the multivariate normal distribution on the x-z plane will be identical to the distribution of the variable x i.e. if we look at the 3D histogram of those points by putting our eye over the x axis the shape of the curve would look like the distribution of x. Same thing with the projection of the curve on the y-z axis.

Exercise 7
Create an histogram of the point in the matrix in the last exercise which the x coordinate are smaller than 1.5 but bigger than 1.3. Then, do the same things for points whose y coordinate are between 10 and 11.

Those are the conditional distributions for some fixed value of x or y. We can see that those conditional distributions are also normally distributed!

Exercise 8
We did before a basic plot of the points from this multivariate distribution, but this plot didn't show the shape of the distribution. We can do better. Use the plot3D package and the hist3D() function (more detail here) to draw the 3d histogram of the dataset of last exercise.

Exercise 9
Another way to represent a 3D distribution in 2D is to use an heatmap. Draw the heatmap of your sample by using:

1. the image2D function from the plot3D package.
2. the hist2d()function from the gplots package.
3. the hexbinplot hexbin package.

Exercise 10
The factor x and y of our multivariate normal distribution are independent, meaning that the value of one value doesn't influence the value of the other. To create a more realistic sample, you should use the mvrnorm package which let you pass a matrix as argument containing the covariance between each variable. This statistics is a measure of the dependence between the factor that take a value between o and 1. You can read more about it here.

Use the mvrnorm function to sample 500 points from a multivariate normal distribution of dimension two. The marginal distribution of the first factor is a normal distribution of mean equal to 5 and a standard deviation of 3,

while the marginal distribution of the second has a mean of 9 and a standard deviation of 1.5. The covariance between both factor is of 0.6.
Then draw the heatmap of this distribution.