



서울대학교
SEOUL NATIONAL UNIVERSITY



서울대학교
데이터사이언스대학원
Graduate School of Data Science
Seoul National University

M3239.003100: Data Analysis and Visualization

Lecture 1

Introduction

Hyunwoo Park
Graduate School of Data Science
Seoul National University

Agenda

- Course Logistics
 - Instructor
 - Grading Scheme
 - Course Policy
- Course Overview
 - Class Schedule
 - Course History and Credit
 - Stata
 - Python
 - Data Collection and Cleaning
 - Static Visualization with Matplotlib
 - Dashboard Design
 - Web Programming
 - Interactive Visual Analysis System with d3.js
 - Why Visualization?
- Class Survey (Due 9/13)
 - Submitted via Google Forms
 - Link to be sent out by this weekend
- Things To Do
 - Install Stata
 - Install the Anaconda distribution of Python 3
 - Launch Jupyter Notebook
 - Get to know the terminal interface
 - Pick a text/code editor

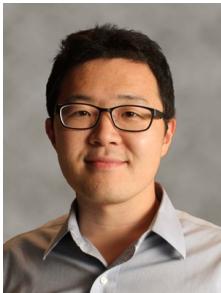
Course Logistics

† Before we begin

- Course syllabus is posted on eTL.
- All course announcements will be posted on eTL.
- Teaching philosophy
 - I will try to make this course relevant to you.
 - I should teach what I believe is actually helpful for you moving forward.
 - I believe in learning by doing, especially for creating something with programming.
 - You should learn from my perspective and organization of thoughts, not just crystalized knowledge.
- Expectation
 - The course is not math heavy but programming heavy.
 - Workload will be very demanding, but I believe you will take away a lot of hands-on skills on data visualization when you complete the course.
 - You are expected to be a self-starter.

Instructor

- About me
 - Associate Professor
Graduate School of Data Science
Seoul National University
 - Email: hyunwoopark@snu.ac.kr
Office: 942-417
Office Hours: By appointment
 - For more information about me, visit
<http://hyunwoopark.com>.
- Before SNU
 - Assistant Professor
Dept. of Operations and Business Analytics
Fisher College of Business
The Ohio State University
Affiliated with Translational Data Analytics
Institute
 - Ph.D. in Industrial Engineering
from Georgia Tech
 - Master in Information Mgt. and Systems
from UC Berkeley
 - B.S. in Electrical Engineering from SNU



.Grading scheme

- Midterm (40%)
- Participation (20%)
 - eTL Activities & Class Survey
- Group Project (40%)
 - Project Proposal (10%)
 - Final Presentation (15%)
 - Output Evaluation (15%)
 - Individual scores may be adjusted based on peer evaluation.

Homework (30%; 6% each)

- No late submission is accepted unless otherwise notified.
- All submissions must be individual work.
- Direct sharing of the code for homework is expressly and strictly prohibited and constitutes an academic misconduct.
- A high-level discussion is permitted and encouraged.

Group project (30%)

- A team of 5-7 people will work together to build an interactive visual analysis system using d3.js.
- I will form groups based on Class Survey AND your requests.
- Please send your group formation request separately. Your request for group formation may not be fully accommodated.
- Your final output may be showcased online.
- Free-riders will be punished by peer evaluation. A student with an extremely and consistently bad peer evaluation may receive a 0 for Group Project, even if the group score is high.
- Deliverables
 - Project proposal (a pdf document)
 - Working interactive visual analysis system
 - System demo video
 - 4+ screenshots highlighting the features of your visual analysis system
 - Peer evaluation

▶ Participation > Class Survey (5%; Due 9/13)

- You will answer survey questions about yourself and your interest.
- It will be conducted using Google Forms.
- Survey answers will be compiled into a dataset to be used for this class.
- Survey link will be sent out by this week.
- Completing this survey properly is 5% of the course grade.
(Some deductions may be applied if instructions were not followed.)

Participation > Online eTL Activities (15%)

- Content-related questions should be first posted on the eTL [Q&A] or [Sharing] board, so that other students can answer for you.
- You earn 1% for each question or answer you post.
- Only questions and answers posted with real name will be counted towards Online Q&A Activities grade.
- You earn credit for maximum one question or answer per day.
- Spammers and trollers may receive a 0, regardless of their past activities.

☒ Grade appeal policy

- All grade appeals should be made within two weeks after the grade for the item is released.
- Final course grade (i.e., letter grade) is not negotiable for any reason.
 - Including, but not limited to, financial burden, job offer deadline, graduation plan, etc.

📅 Office hours policy

- Office hours are offered by appointment only.
- I am unable to offer office hours to give one-on-one handholding sessions.
- You must email me describing briefly why you need a one-on-one conversation with me before scheduling an office hours session.
- Each office hours session is limited by 30 minutes. This is also to avoid a small number of students monopolizing my time on this course.

▶ Language policy

- This course is taught in Korean. All materials will be in English and some Korean.

TA

- Kyu Tae Shim (심규태)
- ktshim@snu.ac.kr

Classroom

- Lectures will be in 942-302 in the following days.
 - 9/1 (Thu)
 - 9/6 (Tue)
 - 9/8 (Thu)
- After that, I will send an announcement on when the class will be in the classroom or just over zoom.
- Even if the class will be in the classroom, it will be broadcasted over zoom.

After all

- I am a new faculty at SNU Graduate School of Data Science who started last year.
- It means this course is still in the making.
- The downside is that there will be inevitably some rough corners and uncertainties here and there.
- The upside is that you will become part of making this course and laying the foundation. I welcome your feedback on how I should design and run this course even during the semester.
- Please be respectful to each other (including me) and I hope all of us enjoy this course this semester.

Course Overview

Brief history of the course

- I developed “Descriptive Analytics and Visualization” course for the Fisher College of Business at The Ohio State University.
- I taught the course for three times, and it became a very popular course there.
- The problem is that the course was developed to teach visualization in R. (By the way, R is also a great choice for data visualization for common charts and building a dashboard. Try it out if you are interested.)

BUSMGT 7331

Descriptive Analytics and Visualization

Professor Hyunwoo Park | Fisher College of Business

A screenshot of a learning management system interface for the course BUSMGT 7331. On the left, there is a vertical sidebar featuring a large image of a modern brick building with many windows, and below it, the book cover for "R for Data Science" by Hadley Wickham & Garrett Grolemund. The main area displays a list of course materials, each represented by a thumbnail image, a title, and a duration. The materials are:

- BM7331 Week1-08: Summary Statistics (16:45)
- BM7331 Week1-07: Types of Variables and Conce... (19:51)
- BM7331 Week1-06: Data Transformation in Tidyv... (20:13)
- BM7331 Week1-05: Tidy Data (23:05)
- BM7331 Week1-04: Data Wrangling Process (12:32)
- BM7331 Week1-03: Metacognitive Strategies (8:07)
- BM7331 Week1-02: Types of Analytics (10:19)
- BM7331 Week1-01: Course Overview (24:35)

Each material entry includes a checkbox, the title, a thumbnail image, the duration, and a "Add description" link.

What students learned in that course

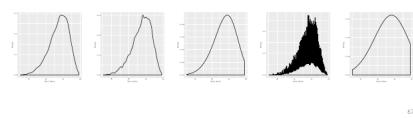
- Topic 1: Data Wrangling and Descriptive Statistics
 - Topic 2: More Data Wrangling, Covariation, and Customizing Visualization
 - Topic 3: Describing Textual Data
 - Topic 4: Geospatial Visualization
 - Topic 5: Network Visualization
 - Topic 6: Interactivity and Dashboard Design
 - Topic 7: Visualizing Models, Dimensionality Reduction, and Clustering

Density plot

- Density plot smooths out and normalizes a histogram

```
with moving window of width bandwidth:
```

```
1 ggplot(df) + geom_density(aes(x=Critic_Score))
2 ggpplot(df) + geom_density(aes(x=Critic_Score), bw=1)
3 ggpplot(df) + geom_density(aes(x=Critic_Score), bw=10)
4 ggpplot(df) + geom_density(aes(x=Critic_Score), adjust=-1)
5 ggpplot(df) + geom_density(aes(x=Critic_Score), adjust=10)
```

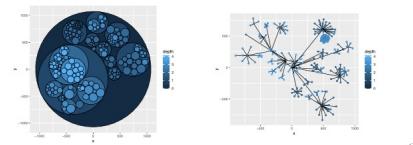


Circlepack layout

```

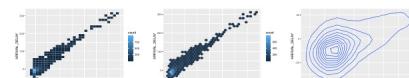
1 ggraph(grf, "circlepack", weight="size") + geom_node_circle(aes(fill =
2 depth), size = 0.25, n = 50) + coord_fixed()
3 ggraph(grf, "circlepack", weight="size") + geom_edge_link() +
4 geom_node_point(aes(color=depth)) + coord_fixed()

```



[3] Counting after binning: bin2d, hex, density_2d

```
1 ggplot(more_filtered_flights) +  
2   geom_bin2d(aes=x=DEPARTURE_DELAY, y=ARRIVAL_DELAY))  
3 ggplot(more_filtered_flights) +  
4   geom_hex(aes=x=DEPARTURE_DELAY, y=ARRIVAL_DELAY))  
5 ggplot(more_filtered_flights) +  
6   geom_density_2d(aes=x=DEPARTURE_DELAY, y=ARRIVAL_DELAY)
```



Word cloud

```
1 library(wordcloud)
2 wc <- tidy_amz %>% count(word, sort=T)
3 wordcloud(wc$word, wc$n, min.freq=50, colors=brewer.pal(5, "Dark2"))
4 wordcloud
```



Interactive data exploration with shinydashboard

The screenshot shows the RStudio IDE with a Shiny application running. The left pane displays the R code for the application, which includes functions for data processing and plotting. The right pane shows the Shiny user interface with two plots: a scatter plot of 'age' vs 'height' and a box plot of 'age'.

```
library(shiny)
library(ggplot2)

# --- Data Processing Functions --#
get_heights_by_year <- function(df) {
  df %>% group_by(year) %>% summarise(height = mean(height))
}

get_boxplot_data <- function(df) {
  df %>% group_by(year) %>% summarise(min = min(height), q1 = quantile(height, 0.25),
                                             median = median(height), q3 = quantile(height, 0.75),
                                             max = max(height))
}

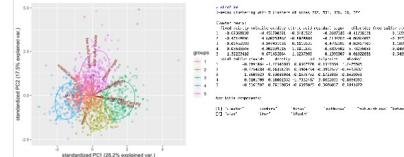
# --- Main Application Function --#
ui <- fluidPage(
  titlePanel("Age vs Height"),
  sidebarLayout(
    sidebarPanel(
      selectInput("year", "Year", choices = c("All", 1950, 1960, 1970, 1980, 1990, 2000, 2010, 2020)),
      selectInput("order", "Order", choices = c("desc", "asc"))
    ),
    mainPanel(
      plotOutput("scatter"),
      plotOutput("boxplot")
    )
  )
)

server <- function(input, output) {
  # Load data
  data <- read_csv("https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/data/2021/2021-01-12/heights.csv")
  
  # Process data
  heights_by_year <- get_heights_by_year(data)
  boxplot_data <- get_boxplot_data(data)
  
  # Create reactive data
  reactive_data <- reactive({
    if (input$year == "All") {
      data
    } else {
      filter(data, year == input$year)
    }
  })
  
  # Create reactive ordering
  reactive_order <- reactive({
    if (input$order == "desc") {
      desc(reactive_data())
    } else {
      reactive_data()
    }
  })
  
  # Create reactive scatter plot data
  reactive_scatter <- reactive({
    reactive_order() %>% ggplot(aes(x = age, y = height)) +
      geom_point()
  })
  
  # Create reactive boxplot data
  reactive_boxplot <- reactive({
    reactive_order() %>% ggplot(aes(x = year, y = height)) +
      geom_boxplot()
  })
  
  # --- Outputs --#
  output$scatter <- renderPlot(reactive_scatter())
  output$boxplot <- renderPlot(reactive_boxplot())
}

shinyApp(ui = ui, server = server)
```

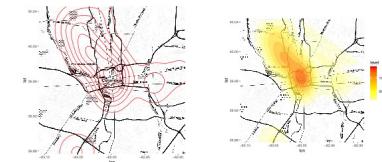
k-means clustering in R

```
1 wine2.km <- kmeans(scale(wine2 %>% select(-highq,-quality)), 5)
2 wine2.km
3 silhouette(wine2.km, alpha=1, measure="fuzzywuzzy/wine2_km$cluster", all=TRUE)
```



Visualizing density with contour and heatmap

```
1 basemap + geom_density2d(data=sbholll2, aes(lon, lat), color="red")
2 basemap +
3 stat_density2d(data=sbholll2, aes(fill=..level..), geom="polygon", alpha=0.5)
4 scale_fill_gradient2(low="white", mid="yellow", high="red", midpoint=0.5)
```



💡 Converting the content for Python and adding d3.js

- Python is a great choice for data visualization if you want more fine-grained customization in your visualization.
- Python is also great for streamlining your data visualization with other computer science or software engineering code.
- Python can also build a web server for an interactive visual analysis system using web programming and d3.js.
- Credits and special thanks go to Georgia Tech Vis Group friends.



Rahul Basole



John Stasko



Alex Endert



Polo Chau



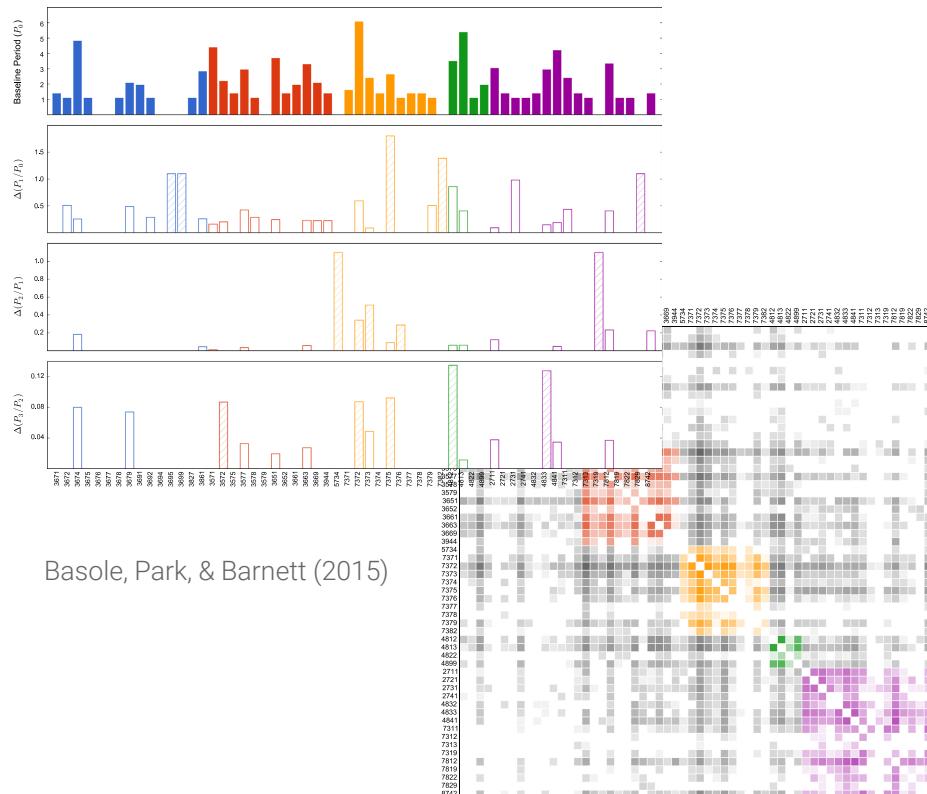
Chad Stolper

▣ Two-track system (back in 2021)

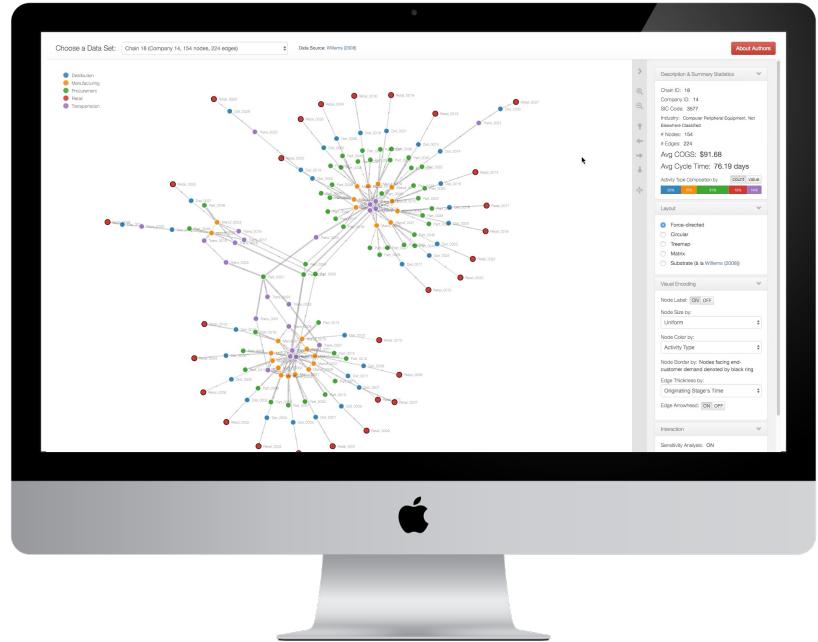
- Python
 - Data collection and cleaning
 - Matplotlib
 - Various high-quality static visualization with customizations
 - Dashboard
 - Evaluation: Homework + Midterm
- d3.js
 - Web programming basics
 - Back-end server with Python
 - Front-end HTML, CSS, JavaScript
 - Front-end framework: Bootstrap
 - d3.js as a JavaScript library
 - Evaluation: Group Project + Midterm

↗ My aspiration

- Python



- d3.js

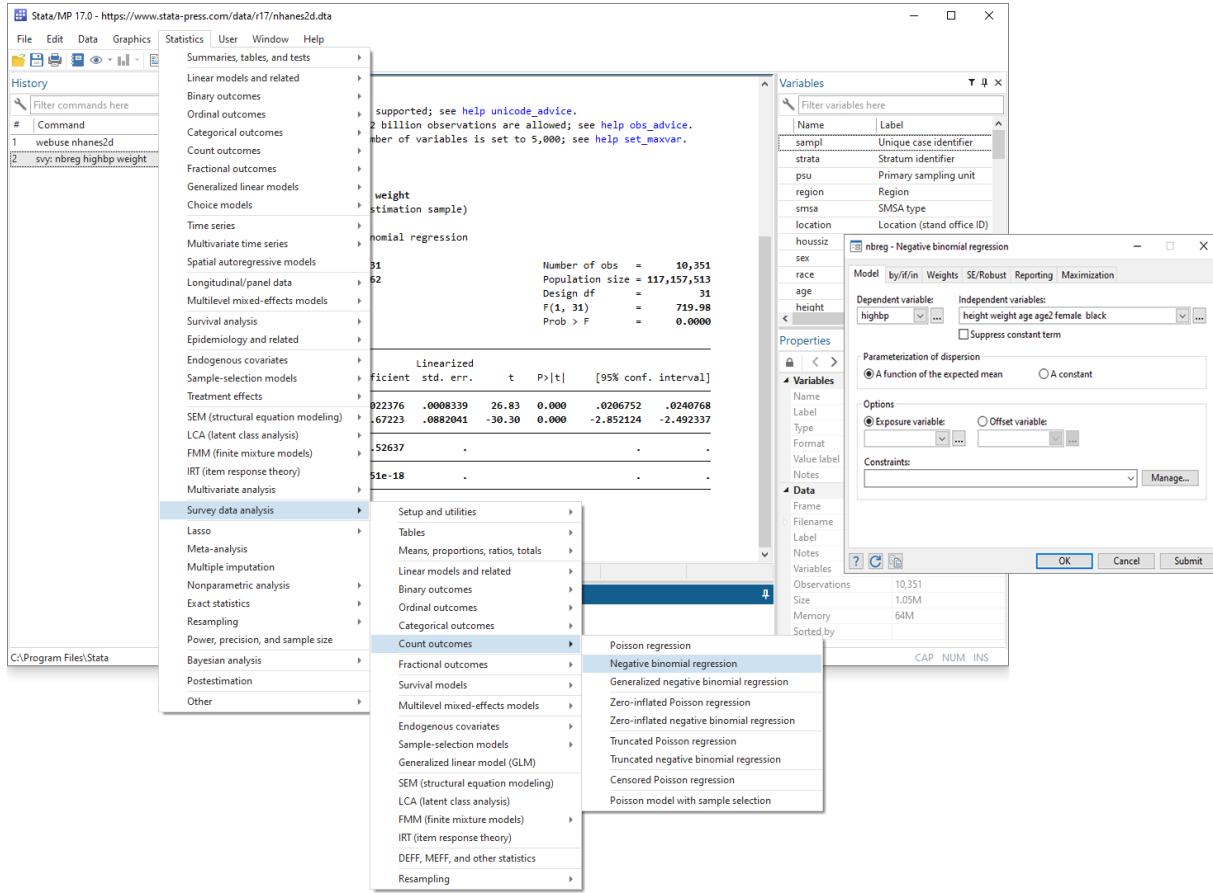


<https://visualscm.herokuapp.com/>

Park, Bellamy, & Basole (2016)

↗ Three-track system (NEW in 2022)

- Stata (New in 2022)
- Python
- D3.js



Things To Do

Install Stata



오류신고

광장 스누인지원 총장추천위원회 캘린더 Dooray

전자결재

정보광장

게시판

스누이벤트

채용/인턴쉽

커뮤니티

SNU NOW

정책연구과제
검색

노트북/SW대여

장비/SW신청

예약목록확인

대여확인/연장신청

연장취소/승인현황

캠퍼스라이선스SW다운

SNU Board
board.snu.ac.kr/api/board/574/11645688777152?boardId=574&cmpBrdId=574&srchKey=&srchType=&srchBgnR...

캠퍼스 라이선스 S/W

※ 다운로드는 서울대학교 내부에서 업무 및 연구, 교육 용도로 사용하는 장치(PC, 노트북 등)에서만 사용 가능.
※ 설치 파일 및 관련 정보를 서울대학교 외부로 유통할 경우 법적 책임을 받으며 설치 및 오류 상당지원 안됨.

Stata 17 SE
IT서비스센터 (정보화지원과) 2022.02.24 (조회수 5,053)

■ 사용 안내

- 이 제품은 서울대학교 내부에서 업무 및 연구, 교육 용도로 사용하는 장치(PC, 노트북 등)에서만 사용 가능.
- 설치 파일 및 관련 정보를 서울대학교 외부로 유통할 경우 법적 책임을 받으며 설치 및 오류 상당지원 안됨.

■ 이용 문의

- 서울대학교 첨단정보본부(02校区) 10층 IT서비스센터
- 전화 번호 : 02-880-8282(AMS-1) (평일 09:00~12:00/13:00~18:00)
- 이메일 : its@snu.ac.kr

■ 제품 정보

- 제품명 : Stata 17 SE
- 소개 : Stata는 확장하고 정확하며, 사용하기 좋습니다. Stata는 여러분의 데이터 과학이 필요로 하는 데이터 관리, 시각화, 통계분석 및 차세대적인 리포트 기능이 제공되는 완벽한 통합 소프트웨어 패키지.
- 시스템 요구 사항 (버전 25) : Windows 8.1/10 32bit/64bit, Mac OS 10.14, Linux

■ 라이선스 (사용권)

- 이 제품은 2023년 3월 1일까지 사용 가능.

■ 설치/설정

- 설치
- 아래 설치 파일에 포함된 설치 안내 문서를 확인 후 설치 진행 비활.

■ 다른문서

- 다운로드는 서울대학교 내부 네트워크(인터넷)을 유선 또는 무선(SNU-1st-time, SNU-Member), 별도 실내 공유기로 연결된 상태에서 가능함.
- ※ 을 클릭하여 버전 확인 경로로 다른문서(저장)다음을 압축 풀기하고 파일 내용을 확인하시기 바랍.
- Windows용 Stata 17 SE
- 설치 파일 (설치 안내 사용권 정보) :
- FAQ (문제 해결) :
- 설치 파일 (설치 안내 사용권 정보) :
- Linux용 Stata 17 SE
- 설치 파일 (설치 안내 사용권 정보) :

※ 위 내용은 첨부된 파일들은 이용 안내, 버전 업데이트, 개선 등의 이유로 계속 수정 또는 교체될 수 있으므로 사용자의 장치(PC, 노트북 등)에서 다운로드 저장하여 보관 중인 오래된 파일을 사용하는 것보다 최근 내용을 확인하여 다시 다운로드한 것으로 사용(설치)를 권장함.

※ 서울대학교 캠퍼스 라이선스 소프트웨어 이용안내 및 불법 소프트웨어 사용 금지 공지 보기 (글자)

첨부파일

목록

이전글 ^ 두레이(Dooray) 메신저 2021.06.07

▶ Install Python 3 Anaconda distribution

- <https://www.anaconda.com/products/individual>
- Python 3.8 (Version 2021.05)
- Included packages:
 - https://docs.anaconda.com/anaconda/packages/py3.8_osx-64/

The screenshot shows the Anaconda Individual Edition landing page. At the top, there's a navigation bar with links for Products, Pricing, Solutions, Resources, Blog, and Company. A prominent green button labeled "Get Started" is on the right. Below the navigation, there's a large green "Q" logo followed by the text "Individual Edition". The main headline reads "Your data science toolkit". A paragraph below it states: "With over 25 million users worldwide, the open-source Individual Edition (Distribution) is the easiest way to perform Python/R data science and machine learning on a single machine. Developed for solo practitioners, it is the toolkit that equips you to work with thousands of open-source packages and libraries." To the right, there's a call-to-action box for "Anaconda Individual Edition" with a "Download" button for MacOS. It also mentions "Python 3.8 • 64-Bit Graphical Installer • 440 MB". At the bottom, there's a link "Get Additional Installers" with icons for Windows, Mac, and Linux.

Launch Jupyter Notebook

The screenshot shows two stacked Jupyter Notebook interfaces.

The top interface is a file browser at `localhost:8888/tree`. It has tabs for "Files", "Running", and "Clusters". Under "Files", there is a list of items:

- A folder icon with "0" files and a "New" button.
- A file icon with "Example Networks.ipynb". Details: "Running" 8 days ago, 441 kB.

The bottom interface is an active notebook at `localhost:8888/notebooks/Untitled.ipynb?kernel_name=python3`. It has a toolbar with File, Edit, View, Insert, Cell, Kernel, Widgets, Help, Trusted, and Python 3 buttons. Below the toolbar is a code cell with the prompt "In []:".

Get to know the terminal interface

```
(base) MacBook-Pro:python_code oksure$ which python
/Users/oksure/opt/anaconda3/bin/python
(base) MacBook-Pro:python_code oksure$ which jupyter
/Users/oksure/opt/anaconda3/bin/jupyter
(base) MacBook-Pro:python_code oksure$ jupyter console
Jupyter console 6.4.0

Python 3.8.8 (default, Apr 13 2021, 12:59:45)
Type 'copyright', 'credits' or 'license' for more information
IPython 7.22.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]:
Do you really want to exit ([y]/n)?
Shutting down kernel
(base) MacBook-Pro:python_code oksure$ [IPKernelApp] WARNING | Parent appears to have exited, shutting down.

(base) MacBook-Pro:python_code oksure$ jupyter notebook
[I 2021-09-02 12:11:37.438 LabApp] JupyterLab extension loaded from /Users/oksure/opt/anaconda3/lib/python3.8/site-packages/jupyterlab
[I 2021-09-02 12:11:37.438 LabApp] JupyterLab application directory is /Users/oksure/opt/anaconda3/share/jupyter/lab
[I 12:11:37.442 NotebookApp] Serving notebooks from local directory: /Users/oksure/Dropbox/Research_SupplyNetwork/python_code
[I 12:11:37.442 NotebookApp] Jupyter Notebook 6.3.0 is running at:
[I 12:11:37.442 NotebookApp] http://localhost:8888/?token=88cbcfda9fdb3afc5d675d6f61b1a7c551e5493321c13a2d
[I 12:11:37.442 NotebookApp] or http://127.0.0.1:8888/?token=88cbcfda9fdb3afc5d675d6f61b1a7c551e5493321c13a2d
[I 12:11:37.442 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 12:11:37.461 NotebookApp]

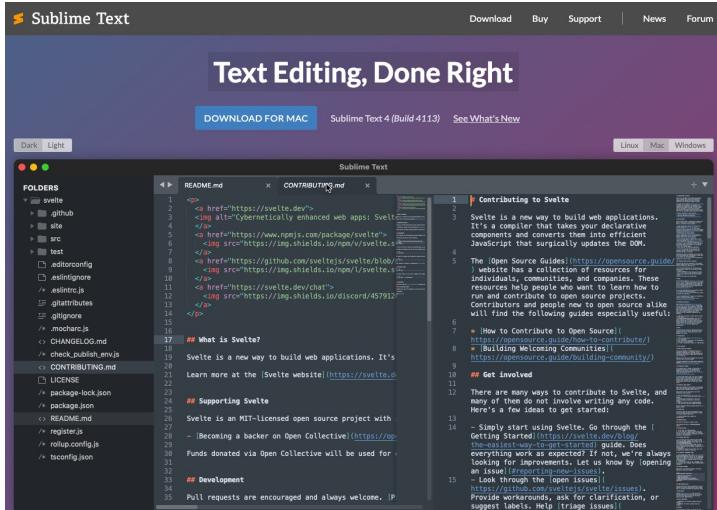
To access the notebook, open this file in a browser:
file:///Users/oksure/Library/Jupyter/runtime/nbserver-61601-open.html
Or copy and paste one of these URLs:
http://localhost:8888/?token=88cbcfda9fdb3afc5d675d6f61b1a7c551e5493321c13a2d
or http://127.0.0.1:8888/?token=88cbcfda9fdb3afc5d675d6f61b1a7c551e5493321c13a2d
```

[0] 1:bash 2:bash- 3:bash 4:python3.8*

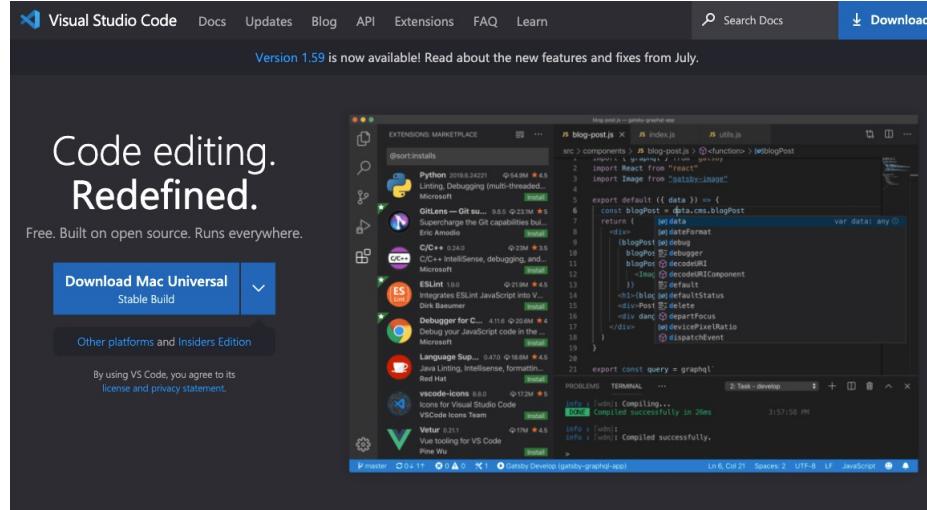
"MacBook-Pro.local" 12:11 02-Sep-21

💡 Pick your code/text editor

- Sublime Text
 - <https://www.sublimetext.com/>



- VS Code
 - <https://code.visualstudio.com/>



Why Visualization?

✚ Why visualization?

- How many 7's are there?

6 1 0 1 5 6 3 8 3 5 3 8 3 2 3 1 8 1 9 1 0 2 1 6 3 3 8 1 6 3 2 9 3 1 8 9 5 0 5 1
4 9 8 9 4 6 5 5 9 5 9 8 5 1 9 3 0 6 1 8 4 0 0 1 4 9 8 3 1 5 9 4 5 3 3 9 3 1 0 2
4 6 1 5 6 3 0 3 6 9 3 7 9 4 8 2 4 8 3 3 4 4 8 6 9 8 0 0 6 0 5 2 1 2 3 9 8 4 1 5
3 3 5 1 5 4 0 1 9 3 3 1 5 7 3 1 9 8 1 5 3 6 9 2 4 2 1 6 5 8 2 5 7 1 5 0 5 6 4 8
9 0 5 9 3 9 4 1 2 2 3 5 2 9 5 9 9 2 3 2 3 6 9 4 2 3 9 0 2 5 3 4 0 8 4 3 5 8 9 8
4 8 1 6 2 6 3 2 3 3 3 3 4 1 9 9 2 2 5 6 4 3 3 2 2 1 9 6 6 4 0 3 9 0 1 2 0 0 2 9
6 3 0 3 2 3 6 0 3 1 2 6 6 3 5 8 2 3 3 3 5 8 0 8 9 9 4 1 2 7 8 3 3 5 6 8 3 4 6 3
4 9 9 0 6 4 3 4 4 5 5 3 0 0 5 3 0 3 5 7 0 6 1 8 0 1 0 0 6 1 2 2 9 8 8 6 6 2 3 6
9 8 1 1 3 5 6 3 8 5 9 4 9 4 2 3 3 1 3 2 6 1 3 6 2 6 8 0 9 3 5 9 8 1 0 4 9 2 9 1
1 0 5 2 2 4 2 0 9 0 3 8 0 3 6 3 3 2 5 4 9 1 4 1 1 4 1 5 8 5 5 3 8 2 3 6 2 2 3 9

▶ Power of visualization

- How about now?

A large grid of numbers from 0 to 9, arranged in approximately 15 rows and 20 columns. The numbers are in a light gray font. Several instances of the digit '7' are highlighted in red. Notable occurrences include a '7' in the 4th row, 3rd column; another in the 5th row, 3rd column; a third in the 7th row, 12th column; and a fourth in the 8th row, 3rd column.

6 1 0 1 5 6 3 8 3 5 3 8 3 2 3 1 8 1 9 1 0 2 1 6 3 3 8 1 6 3 2 9 3 1 8 9 5 0 5 1
4 9 8 9 4 6 5 5 9 5 9 8 5 1 9 3 0 6 1 8 4 0 0 1 4 9 8 3 1 5 9 4 5 3 3 9 3 1 0 2
4 6 1 5 6 3 0 3 6 9 3 7 9 4 8 2 4 8 3 3 4 4 8 6 9 8 0 0 6 0 5 2 1 2 3 9 8 4 1 5
3 3 5 1 5 4 0 1 9 3 3 1 5 7 3 1 9 8 1 5 3 6 9 2 4 2 1 6 5 8 2 5 7 1 5 0 5 6 4 8
9 0 5 9 3 9 4 1 2 2 3 5 2 9 5 9 9 2 3 2 3 6 9 4 2 3 9 0 2 5 3 4 0 8 4 3 5 8 9 8
4 8 1 6 2 6 3 2 3 3 3 3 4 1 9 9 2 2 5 6 4 3 3 2 2 1 9 6 6 4 0 3 9 0 1 2 0 0 2 9
6 3 0 3 2 3 6 0 3 1 2 6 6 3 5 8 2 3 3 3 5 8 0 8 9 9 4 1 2 7 8 3 3 5 6 8 3 4 6 3
4 9 9 0 6 4 3 4 4 5 5 3 0 0 5 3 0 3 5 7 0 6 1 8 0 1 0 0 6 1 2 2 9 8 8 6 6 2 3 6
9 8 1 1 3 5 6 3 8 5 9 4 9 4 2 3 3 1 3 2 6 1 3 6 2 6 8 0 9 3 5 9 8 1 0 4 9 2 9 1
1 0 5 2 2 4 2 0 9 0 3 8 0 3 6 3 3 2 5 4 9 1 4 1 1 4 1 5 8 5 5 3 8 2 3 6 2 2 3 9

↗ Power of human vision

- Unparalleled information processing capacity compared to other senses
 - Sight: 1250MB/s
 - Touch: 125MB/s
 - Hearing / smelling: 12.5MB/s
- https://www.ted.com/talks/david_mccandless_the_beauty_of_data_visualization
or
<https://youtu.be/pLqjQ55tz-U>
- An article about his talk:
<https://www.theguardian.com/science/punctuated-equilibrium/2010/dec/30/2>



Details

About the talk

Transcript

31 languages

Comments (289)

Join the conversation

David McCandless turns complex data sets (like worldwide military spending, media buzz, Facebook status updates) into beautiful, simple diagrams that tease out unseen patterns and connections. Good design, he suggests, is the best way to navigate information glut -- and it may just change the way we see the world.

This talk was presented at an official TED conference, and was featured by our editors on the home page.

ABOUT THE SPEAKER



David McCandless · Data journalist

David McCandless draws beautiful conclusions from complex datasets -- thus revealing unexpected insights into our world.

2,843,959 views

TEDGlobal 2010 | July 2010

Related tags

Complexity
Computers
Data

💡 Trivia about the trivia

- I created these random numbers using Python.

```
1 import random
2 a = random.choices(range(10), k=400)
3 for i in range(10): print(' '.join(str(v) for v in a[i*40:(i+1)*40]))
```

```
In [6]: for i in range(10): print(' '.join(str(v) for v in a[i*40:(i+1)*40]))
5 6 1 0 9 9 2 2 8 4 6 0 1 2 6 7 8 3 8 4 3 2 6 4 9 7 9 7 5 3 4 8 9 0 0 9 4 2 6 0
6 8 5 1 4 6 6 8 9 5 7 2 0 5 6 5 3 0 7 5 1 4 7 3 8 6 7 5 9 3 7 9 5 9 4 6 6 4 8 4
0 0 2 6 5 7 7 1 8 4 9 4 7 3 9 7 9 4 4 1 8 0 9 0 1 8 3 6 0 3 0 8 6 7 1 3 3 7 4 6
6 2 7 4 3 1 9 5 2 9 8 8 6 1 4 2 7 4 6 9 6 8 0 8 8 2 2 3 2 8 1 9 1 9 7 3 9 8 9 4
7 9 1 8 7 5 8 8 6 1 1 4 3 8 5 4 6 3 1 4 5 9 3 4 4 7 8 1 5 2 0 8 1 2 1 2 2 5 9 5
5 8 7 6 0 0 7 2 9 3 2 8 5 6 6 7 2 0 1 9 0 2 0 7 8 5 0 7 1 4 9 5 5 6 3 5 8 4 9 7
7 9 2 1 3 2 0 7 2 5 0 2 5 3 0 6 9 0 9 8 8 3 7 7 2 3 7 6 0 4 4 8 4 8 2 7 4 4 5 8
9 2 4 6 9 9 4 4 7 3 3 1 9 4 0 7 0 3 4 5 8 5 8 2 9 0 4 8 7 4 1 4 3 4 0 7 7 6 6 5
1 8 2 8 8 2 6 4 5 8 2 7 7 7 6 3 2 9 7 3 5 9 7 4 6 1 9 6 3 8 2 4 4 0 4 4 7 4 4 3
6 0 0 9 6 0 0 9 1 6 4 9 8 8 4 0 8 1 3 1 0 5 1 0 3 3 2 5 7 7 9 5 8 9 6 9 2 7 5 4
```

💡 How did I figure this out?

- I didn't solve it right away. So I Googled.
- <https://docs.python.org/3/library/random.html>

A screenshot of a Google search results page. The search query "python random sample" is entered in the search bar. Below the search bar, there are filters for "All", "Images", "Videos", "News", "Books", and "More". The results section shows a snippet from the Python documentation for the `random` module. The snippet includes the title "random — Generate pseudo-random numbers — Python", a brief description, and a "Table of Contents" sidebar. The main content area continues with the module's documentation, mentioning its implementation of pseudo-random number generators and various distribution functions.

python random sample

All Images Videos News Books More

About 79,700,000 results (0.75 seconds)

<https://docs.python.org> › library › random

random — Generate pseudo-random numbers — Python

Almost all module functions depend on the basic function `random()`, which generates floating point numbers in the range [0.0, 1.0]. Used for random sampling without replacement.

People also search for

- [python random sample from dataframe](#)
- [python random int](#)
- [python random sample from list](#)
- [python random sample with replacement](#)
- [python random choices](#)
- [python random number](#)

Table of Contents

- random — Generate pseudo-random numbers
 - Bookkeeping functions
 - Functions for bytes
 - Functions for integers
 - Functions for sequences
 - Real-valued distributions
 - Alternative Generator
 - Notes on Reproducibility
 - Examples
 - Recipes

random — Generate pseudo-random numbers

Source code: [Lib/random.py](#)

This module implements pseudo-random number generators for various distributions.

For integers, there is uniform selection from a range. For sequences, there is uniform selection of a random element, a function to generate a random permutation of a list in-place, and a function for random sampling without replacement.

On the real line, there are functions to compute uniform, normal (Gaussian), lognormal, negative exponential, gamma, and beta distributions. For generating distributions of angles, the von Mises distribution is available.

Not quite there yet... And finally!

- What I wanted is to randomly sample integers between 0 and 9 “with replacement.”
- random.sample is sampling without replacement.
- I kept search the documentation and found there's another function called “choices.”

```
>>> sample([10, 20, 30, 40, 50], k=4)      # Four samples without replacement
[40, 10, 50, 30]
```

Simulations:

```
>>> # Six roulette wheel spins (weighted sampling with replacement)
>>> choices(['red', 'black', 'green'], [18, 18, 2], k=6)
['red', 'green', 'black', 'black', 'red', 'black']
```

💡 Metacognition

- Meaning
 - Thinking about own thinking
 - Knowing about own knowing
 - Awareness of one's awareness
- Metacognitive knowledge
 - (1) Content knowledge (declarative knowledge): Knowing what you know and what you don't know
 - (2) Task knowledge (procedural knowledge): Knowing the procedure to find a solution for the problem
 - (3) Strategic knowledge (conditional knowledge): Knowing what to do to learn how to solve the problem
- Articulating what you do NOT know is paramount
in accelerating self-learning in the era of Googling everything.

❸ Key Takeaway

- Knowing what to search for is really really important.
- Google is your friend. Stack Overflow is usually a Google's go-to friend.
- Python documentation is another Google's friend.

Thank you!