Portofolio Data Science

Projek dengan DATACAMP: Survei Ilmu Data Kaggle (R)

Ketika memulai karir dalam ilmu data, orang sering bertanya-tanya alat dan bahasa pemrograman apa yang digunakan di industri, dan keterampilan apa yang harus dipelajari terlebih dahulu. Dengan menjelajahi hasil Survei Ilmu Data Kaggle 2017, terdapat alat yang digunakan oleh 10.000+ orang dalam komunitas ilmu data profesional.

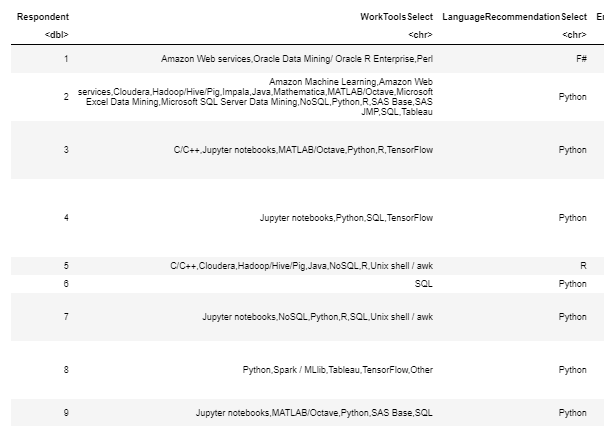
Project ini menggunakan subset dari dataset Kaggle Machine Learning dan Data Science Survey 2017.

1. Ilmu Data  
 Di seluruh dunia ilmu data, ada banyak bahasa dan alat yang dapat digunakan untuk menyelesaikan tugas yang diberikan. Mempelajari apa yang digunakan oleh para profesional di industri ilmu data saat bekerja dapat membantu Anda mendapatkan pemahaman yang lebih baik tentang hal-hal yang mungkin akan diminta untuk Anda lakukan di masa mendatang. Dalam proyek ini, saya mencari tahu alat dan bahasa apa yang digunakan para profesional dalam pekerjaan mereka sehari-hari. Data berasal dari Survei Ilmu Data Kaggle yang mencakup tanggapan dari lebih dari 10.000 orang yang menulis kode untuk menganalisis data dalam pekerjaan sehari-hari mereka.

# Load necessary packages  
library(tidyverse)

# Load the data  
responses <- read\_csv(‘datasets/kagglesurvey.csv’)

# Print the first 10 rows  
head(responses, n = 10)



output

2. Menggunakan berbagai alat  
 Sekarang setelah memuat hasil survei, Untuk mendapatkan gambaran yang lebih baik tentang bagaimana data diformat, saya akan melihat penggunaan alat responden pertama dan melihat bahwa pengambil survei ini membuat daftar beberapa alat yang masing-masing dipisahkan oleh koma. Untuk mempelajari berapa banyak orang yang menggunakan setiap alat, kita perlu memisahkan semua alat yang digunakan oleh masing-masing individu. Ada beberapa cara untuk menyelesaikan tugas ini, tetapi kita akan menggunakan str\_split () dari stringr untuk memisahkan alat di setiap koma. Karena itu akan membuat daftar di dalam bingkai data, kita bisa menggunakan fungsi tidyr unest () untuk memisahkan setiap item daftar ke baris baru.

# Printing the first respondent’s tools and languages  
responses[1, 2]

# Add a new column, and unnest the new column  
tools <- responses %>%  
mutate(work\_tools = str\_split(WorkToolsSelect, “,”)) %>%  
unnest(work\_tools)

# View the first 6 rows of tools  
head(tools)



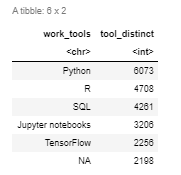
output

3. Menghitung pengguna tiap alat  
Sekarang setelah memisahkan semua alat yang digunakan oleh setiap responden, kita dapat mengetahui alat mana yang paling populer.

# Creating a new data frame  
tool\_count <- tools

# Grouping the data by work\_tools, calculate the number of responses in each group  
tool\_count <- tool\_count %>%  
group\_by(work\_tools) %>%  
summarise(tool\_distinct = n()) %>%  
arrange(desc(tool\_distinct))

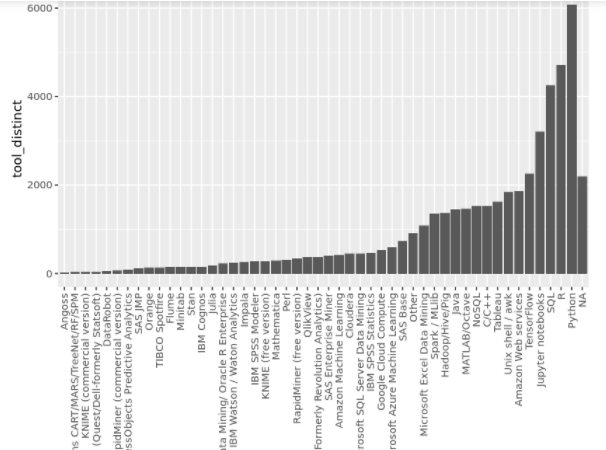
# Printing the first 6 results  
head(tool\_count, 6)



output

4. Merencanakan alat yang paling populer

# Creating a bar chart of the work\_tools column.  
# Arranging the bars so that the tallest are on the far right  
ggplot(tool\_count, aes(x = reorder(work\_tools, tool\_distinct), y = tool\_distinct)) +  
geom\_bar(stat = “identity”) +  
theme(axis.text.x = element\_text(angle = 90, vjust = .5, hjust = 1))

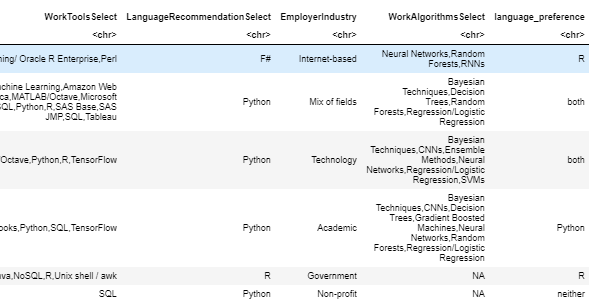


output

5. Perdebatan R vs Python  
Dalam bidang ilmu data, ada banyak perdebatan di antara para profesional tentang apakah R atau Python harus menjadi yang tertinggi. Pada gambar terakhir terlihat bahwa R dan Python adalah dua bahasa yang paling umum digunakan, tetapi mungkin saja banyak responden yang menggunakan R dan Python.

# Filter for China, from Feb 15  
china\_after\_feb15 <- confirmed\_cases\_china\_vs\_world %>%  
 filter(is\_china == “China”, date >= “2020–02–15”)

# Using china\_after\_feb15, draw a line plot cum\_cases vs. date  
# Add a smooth trend line using linear regression, no error bars  
ggplot(china\_after\_feb15, aes(date, cum\_cases)) +  
 geom\_line() +  
 geom\_smooth(method = “lm”, se = FALSE) +  
 ylab(“Cumulative confirmed cases”)



output

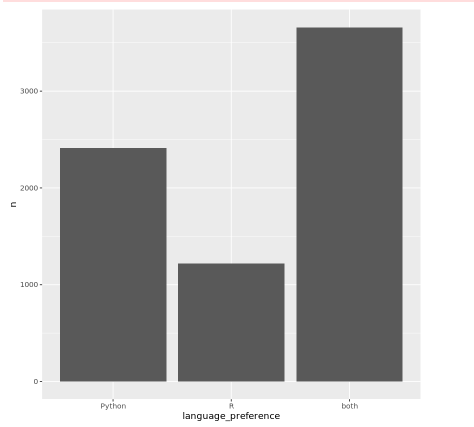
6. Merencanakan pengguna R vs Python  
Sekarang hanya perlu melihat lebih dekat berapa banyak responden yang menggunakan R, Python, dan keduanya!

# Creating a new data frame  
debate\_plot <- debate\_tools

# Grouping by language preference and calculate number of responses  
debate\_plot <- debate\_plot %>%  
group\_by(language\_preference) %>%  
summarise(n = n()) %>%

# Removing the row for users of “neither”  
filter(language\_preference != “neither”)

# Creating a bar chart  
ggplot(debate\_plot, aes(x = language\_preference, y = n))+  
geom\_bar(stat = “identity”)



output

7. Rekomendasi bahasa  
Sepertinya grup program profesional terbesar di Python dan R. Tapi apa yang terjadi ketika mereka ditanya bahasa mana yang mereka rekomendasikan untuk pelajar baru? Apakah pecinta R selalu merekomendasikan R?

# Creating a new data frame  
recommendations <- debate\_tools

# Grouping by language\_preference and then LanguageRecommendationSelect  
recommendations <- recommendations %>%  
group\_by(language\_preference, LanguageRecommendationSelect) %>%  
summarise(n = n()) %>%

# Removing empty responses and include the top recommendations  
filter(LanguageRecommendationSelect != 0) %>%  
arrange(desc(LanguageRecommendationSelect)) %>%  
mutate(count = row\_number()) %>%  
filter(count <= 4)

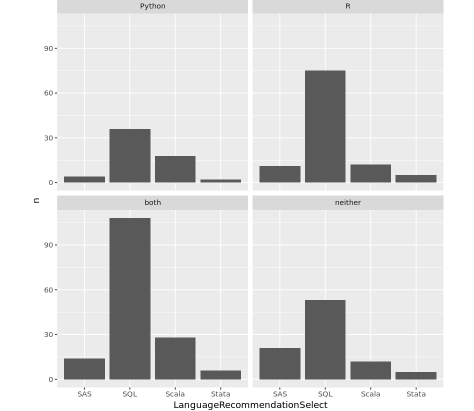
8. Bahasa yang paling direkomendasikan menurut bahasa yang digunakan

Tinggal satu hal lagi. Menentukan secara grafis bahasa mana yang paling direkomendasikan berdasarkan bahasa yang digunakan seseorang.

# Run this to get the data for each country  
confirmed\_cases\_by\_country <- read\_csv(“datasets/confirmed\_cases\_by\_country.csv”)  
glimpse(confirmed\_cases\_by\_country)

# Group by country, summarize to calculate total cases, find the top 7  
top\_countries\_by\_total\_cases <- confirmed\_cases\_by\_country %>%  
 group\_by(country) %>%  
 summarize(total\_cases = max(cum\_cases)) %>%  
 top\_n(7, total\_cases)

# See the result  
top\_countries\_by\_total\_cases



output

9. Moral cerita  
Jadi kami telah berhasil sampai akhir. Kami telah menemukan bahwa Python adalah bahasa paling populer yang digunakan di antara data scientist Kaggle, tetapi pengguna R juga tidak ketinggalan. Dan sementara pengguna Python mungkin sangat menyarankan agar pelajar baru mempelajari Python, akankah pengguna R menemukan pernyataan berikut BENAR atau SALAH?

# Would R users find this statement TRUE or FALSE?  
R\_is\_number\_one = TRUE

# Era Baru Analisis Data dalam Baseball

# Sumber Project : DATACAMP

# Ada era baru analisis data dalam bisbol. Menggunakan teknologi baru yang disebut Statcast, Major League Baseball sekarang mengumpulkan lokasi dan pergerakan yang tepat dari bola dan pemainnya. Dalam proyek ini, saya menggunakan data Statcast untuk membandingkan home run dari dua bintang bisbol paling cemerlang (dan terbesar), Aaron Judge (6'7 ") dan Giancarlo Stanton (6'6"), keduanya sekarang bermain untuk New York Yankees.

# Dataset yang digunakan dalam proyek ini berasal dari Baseball Savant.

1. Revolusi St*a*tcast



Ini adalah Aaron Judge. Judge adalah salah satu pemain terbesar secara fisik di Major League Baseball dengan tinggi 6 kaki 7 inci (2,01 m) dan berat 282 pound (128 kg). Dia juga melakukan home run terberat yang pernah tercatat. Bagaimana kita mengetahui hal ini? Statcast.Statcast adalah sistem pelacakan mutakhir yang menggunakan kamera resolusi tinggi dan peralatan radar untuk mengukur lokasi dan pergerakan yang tepat dari pemain bola dan bisbol. Diperkenalkan pada tahun 2015 untuk semua 30 pertandingan liga utama, data Statcast merevolusi permainan. Tim terlibat dalam “perlombaan senjata” analisis data, mempekerjakan analis kiri dan kanan dalam upaya untuk mendapatkan keunggulan dalam persaingan mereka. Video yang menggambarkan sistem ini luar biasa. Dalam buku catatan ini, kita akan membahas, menganalisis, dan memvisualisasikan data Statcast untuk membandingkan Tuan Hakim dan rekan satu timnya (yang sangat besar). Mari kita mulai dengan memuat data ke dalam Notebook kita. Ada dua file CSV, jud.csv dan stanton.csv, keduanya berisi data Statcast 2015–2017. Kami akan menggunakan DataFrames pandas untuk menyimpan data ini. Mari kita juga memuat perpustakaan visualisasi data kita, matplotlib dan seaborn.

import pandas as pd  
import matplotlib.pyplot as plt  
import seaborn as sns  
%matplotlib inline

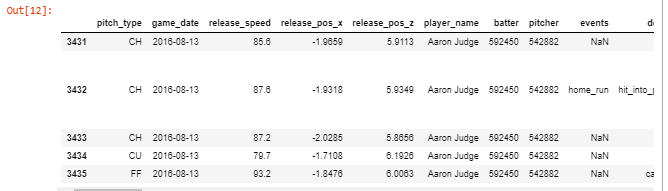
# Load Aaron Judge’s Statcast data  
judge = pd.read\_csv(‘datasets/judge.csv’)

# Load Giancarlo Stanton’s Statcast data  
stanton = pd.read\_csv(‘datasets/stanton.csv’)

2. Apa yang dapat diukur oleh Statcast? Pertanyaan yang lebih baik mungkin, apa yang tidak bisa diukur oleh Statcast? Dimulai dengan pelempar, Statcast dapat mengukur titik data sederhana seperti kecepatan. Pada saat yang sama, Statcast menggali jauh lebih dalam, juga mengukur titik pelepasan dan kecepatan putaran setiap nada. Pindah ke pemukul, Statcast mampu mengukur kecepatan keluar, sudut peluncuran dan vektor bola saat lepas dari bat. Dari sana, Statcast juga dapat melacak waktu hang dan jarak proyeksi yang dilalui bola. Mari kita periksa lima baris terakhir dari DataFrame juri. Anda akan melihat bahwa setiap baris mewakili satu lemparan yang dilemparkan ke pemukul. Anda juga akan melihat bahwa beberapa kolom memiliki nama esoterik. Jika ini tidak masuk akal sekarang, jangan khawatir. Yang relevan akan dijelaskan seperlunya.

# Display all columns (pandas will collapse some columns if we don’t set this option)  
pd.set\_option(‘display.max\_columns’, None)

# Display the last five rows of the Aaron Judge file  
judge.tail()



Output

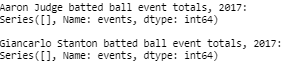
3. Aaron Judge dan Giafncarlo Stanton, pemalas yang produktif



Ini Giancarlo Stanton. Dia juga manusia yang sangat besar, dengan tinggi 6 kaki 6 inci dan berat 245 pon. Meski tidak mengenakan jersey yang sama dengan Judge pada gambar yang diberikan, di tahun 2018 ini mereka menjadi rekan satu tim di New York Yankees. Mereka serupa dalam banyak hal, salah satunya karena mereka melakukan banyak home run. Stanton dan Judge memimpin baseball di home run pada tahun 2017, dengan masing-masing 59 dan 52. Ini adalah total yang luar biasa — pemain di urutan ketiga “hanya” melakukan 45 home run. Stanton dan Judge juga berbeda dalam banyak hal. Salah satunya adalah acara bola yang dipukul, yaitu setiap bola yang dipukul yang menghasilkan suatu hasil. Ini termasuk out, hit, dan error. Selanjutnya, Anda akan menemukan jumlah pertandingan bola memukul untuk setiap pemain pada tahun 2017. Frekuensi acara lainnya sangat berbeda.

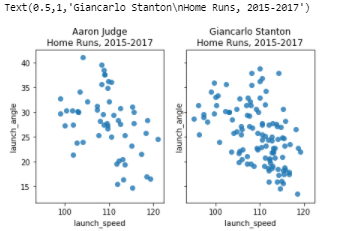
# All of Aaron Judge’s batted ball events in 2017  
judge\_events\_2017 = judge.loc[judge[‘events’] == 2017].events  
print(“Aaron Judge batted ball event totals, 2017:”)  
print(judge\_events\_2017.value\_counts())

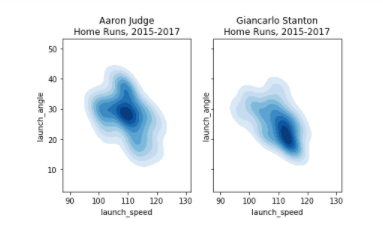
# All of Giancarlo Stanton’s batted ball events in 2017  
stanton\_events\_2017 = stanton.loc[stanton[‘events’] == 2017].events  
print(“\nGiancarlo Stanton batted ball event)



Output

4. Menganalisis home run dengan data Statcast  
Dua dari metrik Statcast yang paling inovatif adalah sudut peluncuran dan kecepatan keluar.Sudut peluncuran: sudut vertikal di mana bola meninggalkan pemukul pemain  
Exit velocity: kecepatan bola bisbol saat melepaskan bat  
Data baru ini telah mengubah cara tim menilai baik hitter maupun pitcher. Mengapa? Sesuai dengan Washington Post:Bola yang dipukul dengan sudut peluncuran tinggi lebih cenderung menghasilkan pukulan.



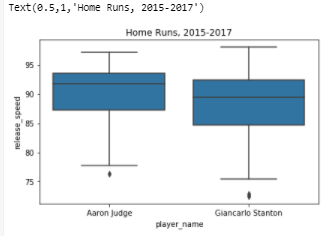


Output

5. Home run dengan pitch velocity  
Statcast tidak hanya mengukur kecepatan bola yang keluar dari pemukul, tetapi juga mengukur kecepatan bola yang keluar dari tangan pelempar dan memulai perjalanannya menuju plate. Kita dapat menggunakan data ini untuk membandingkan home run Stanton dan Judge dalam hal kecepatan nada. Selanjutnya Anda akan menemukan plot kotak yang menampilkan ringkasan lima angka untuk setiap pemain: minimum, kuartil pertama, median, kuartil ketiga, dan maksimum.

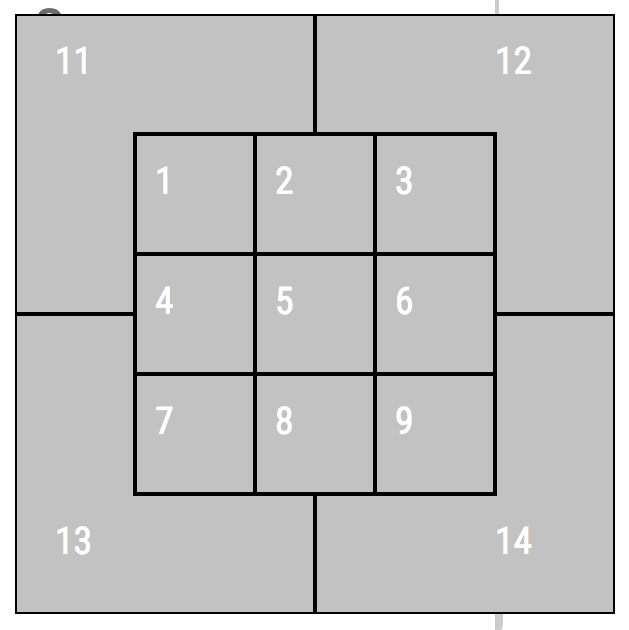
# Combine the Judge and Stanton home run DataFrames for easy boxplot plotting  
judge\_stanton\_hr = pd.concat([judge\_hr, stanton\_hr])

# Create a boxplot that describes the pitch velocity of each player’s home runs  
sns.boxplot(x=’player\_name’,y=’release\_speed’, color=’tab:blue’, data=judge\_stanton\_hr).set\_title(‘Home Runs, 2015–2017’)



Output

6. Tim tuan rumah berdasarkan lokasi lapangan (I)  
Jadi Judge tampaknya memukul rumahnya dengan lemparan yang lebih cepat daripada Stanton. Kita mungkin menyebut Judge sebagai pemukul fastball. Stanton tampak agnostik terhadap kecepatan nada dan kemungkinan pergerakan pitchkarena nada yang lebih lambat (misalnya bola melengkung, penggeser, dan pergantian) cenderung memiliki lebih banyak jeda. Statcast melacak pergerakan dan jenis nada, tetapi mari beralih ke hal lain: lokasi nada. Statcast melacak zona lapangan saat melintasi pelat. Penomoran zona terlihat seperti ini (dari sudut pandang penangkap):



def assign\_x\_coord(row):  
“””  
Assigns an x-coordinate to Statcast’s strike zone numbers. Zones 11, 12, 13,  
and 14 are ignored for plotting simplicity.  
“””  
# Left third of strike zone  
if row.zone in [1, 4, 7]:  
return 1  
# Middle third of strike zone  
if row.zone in [2, 5, 8]:  
return 2  
# Right third of strike zone  
if row.zone in [3, 6, 9]:  
return 3

7. Tim tuan rumah berdasarkan lokasi lapangan (II)  
Dan mari lakukan hal yang sama tetapi untuk koordinat y.

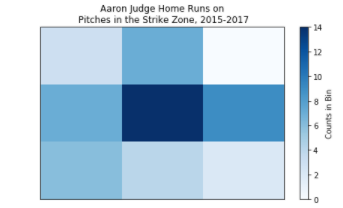
def assign\_y\_coord(row):  
“””  
Assigns a y-coordinate to Statcast’s strike zone numbers. Zones 11, 12, 13,  
and 14 are ignored for plotting simplicity.  
“””  
# Upper third of strike zone  
if row.zone in [1,2,3]:  
return 3  
# Middle third of strike zone  
if row.zone in [4,5,6]:  
return 2  
# Lower third of strike zone  
if row.zone in [7,8,9]:  
return 1

8. Zona home run Aaron Judge  
Sekarang kita dapat menerapkan fungsi yang telah kita buat kemudian membuat histogram 2D kita. Pertama, untuk Aaron Judge (sekali lagi, untuk lemparan di strike zone yang menghasilkan home run).

judge\_strike\_hr = judge\_hr.copy().loc[judge\_hr.zone <= 9]

# Assign Cartesian coordinates to pitches in the strike zone for Judge home runs  
judge\_strike\_hr[‘zone\_x’] = judge\_strike\_hr.apply(assign\_x\_coord, axis=1)  
judge\_strike\_hr[‘zone\_y’] = judge\_strike\_hr.apply(assign\_y\_coord, axis=1)

# Plot Judge’s home run zone as a 2D histogram with a colorbar  
plt.hist2d(judge\_strike\_hr[‘zone\_x’], judge\_strike\_hr[‘zone\_y’], bins = 3, cmap=’Blues’)  
plt.title(‘Aaron Judge Home Runs on\n Pitches in the Strike Zone, 2015–2017’)  
plt.gca().get\_xaxis().set\_visible(False)  
plt.gca().get\_yaxis().set\_visible(False)  
cb = plt.colorbar()



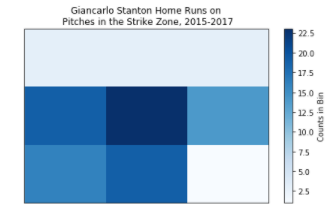
Output

9. Zona home run Giancarlo Stanton  
Dan sekarang untuk Giancarlo Stanton.

# Zones 11, 12, 13, and 14 are to be ignored for plotting simplicity  
stanton\_strike\_hr = stanton\_hr.copy().loc[stanton\_hr.zone <= 9]

# Assign Cartesian coordinates to pitches in the strike zone for Stanton home runs  
stanton\_strike\_hr[‘zone\_x’] = stanton\_strike\_hr.apply(assign\_x\_coord, axis=1)  
stanton\_strike\_hr[‘zone\_y’] = stanton\_strike\_hr.apply(assign\_y\_coord, axis=1)

# Plot Stanton’s home run zone as a 2D histogram with a colorbar  
plt.hist2d(stanton\_strike\_hr[‘zone\_x’], stanton\_strike\_hr[‘zone\_y’], bins = 3, cmap=’Blues’)  
plt.title(‘Giancarlo Stanton Home Runs on\n Pitches in the Strike Zone, 2015–2017’)  
plt.gca().get\_xaxis().set\_visible(False)  
plt.gca().get\_yaxis().set\_visible(False)  
cb = plt.colorbar()  
cb.set\_label(‘Counts in Bin’)



Output

10. Haruskah pelempar lawan takut?  
Beberapa kesimpulan:

Stanton tidak melakukan banyak home run di lapangan di sepertiga bagian atas zona serangan.  
Seperti hampir semua hitter, kedua pemain menyukai lemparan di tengah plate horizontal dan vertikal.  
Lemparan home run paling tidak favorit dari juri tampaknya jauh sementara Stanton tampaknya jauh.  
Jika kita mendeskripsikan zona home run Stanton, itu akan berada di tengah-dalam. Zona home run juri jauh lebih tersebar.  
Pengambilan utama dari keseluruhan latihan ini: Aaron Judge dan Giancarlo Stanton tidak identik meskipun mereka memiliki kemiripan yang dangkal. Dalam hal home run, profil peluncuran mereka, serta kecepatan pitch dan preferensi lokasinya, berbeda.

Haruskah pelempar lawan masih takut?

​# Should opposing pitchers be wary of Aaron Judge and Giancarlo Stanton  
should\_pitchers\_be\_scared = True

Risiko dan Pengembalian: Rasio Sharpe by Projek DATACAMP



1.William Sharpe  
 Investasi mungkin masuk akal jika kita mengharapkannya menghasilkan lebih banyak uang daripada biayanya. Tetapi hasil hanya sebagian dari cerita karena berisiko — mungkin ada berbagai kemungkinan hasil. Bagaimana seseorang membandingkan investasi yang berbeda yang mungkin memberikan hasil yang sama secara rata-rata, tetapi menunjukkan tingkat risiko yang berbeda? Masukkan William Sharpe. Dia memperkenalkan rasio reward-to-variability pada tahun 1966 yang kemudian disebut Rasio Sharpe. Ini membandingkan pengembalian yang diharapkan untuk dua peluang investasi dan menghitung pengembalian tambahan per unit risiko yang dapat diperoleh investor dengan memilih salah satu dari yang lain. Secara khusus, ini melihat perbedaan pengembalian untuk dua investasi dan membandingkan perbedaan rata-rata dengan standar deviasi (sebagai ukuran risiko) dari perbedaan ini. Rasio Sharpe yang lebih tinggi berarti imbalannya akan lebih tinggi untuk sejumlah risiko tertentu. Merupakan hal yang umum untuk membandingkan peluang tertentu dengan tolok ukur yang mewakili seluruh kategori investasi.

Rasio Sharpe telah menjadi salah satu ukuran risiko / pengembalian paling populer di bidang keuangan, paling tidak karena sangat mudah digunakan. Ini juga membantu Profesor Sharpe memenangkan Hadiah Nobel Ekonomi pada tahun 1990 untuk karyanya pada model penetapan harga aset modal (CAPM).Mari pelajari tentang rasio Sharpe dengan menghitungnya untuk saham dua raksasa teknologi Facebook dan Amazon. Sebagai patokan, kami akan menggunakan S&P 500 yang mengukur kinerja 500 saham terbesar di AS.

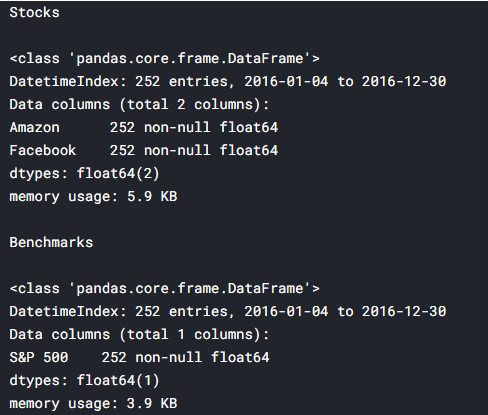
import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt

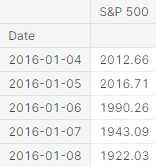
# Settings to produce nice plots in a Jupyter notebook  
plt.style.use(‘fivethirtyeight’)  
%matplotlib inline

# Reading in the data  
stock\_data = pd.read\_csv(‘../input/stock\_data.csv’,  
parse\_dates=[‘Date’],  
index\_col=[‘Date’]).dropna()

benchmark\_data = pd.read\_csv(‘../input/benchmark\_data.csv’,  
parse\_dates=[‘Date’],  
index\_col=[‘Date’]).dropna()

1. Sekilas data Mari kita lihat datanya untuk mengetahui berapa banyak observasi dan variabel yang kita miliki.

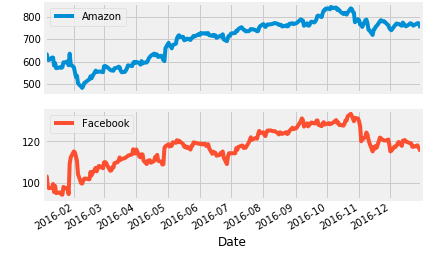




Output

3. Plot & rangkum harga harian untuk Amazon dan Facebook  
Sebelum kita membandingkan investasi di Facebook atau Amazon dengan indeks 500 perusahaan terbesar di AS, mari kita visualisasikan datanya, jadi kita lebih memahami apa yang kita hadapi.  
*# visualize the stock\_data*  
stock\_data.plot(subplots=True)  
plt\_title = ‘Stock Data’  
plt.show()

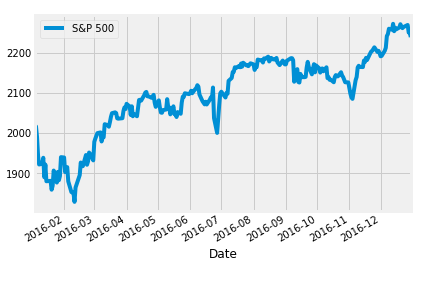
*# summarize the stock\_data*  
stock\_data.describe()

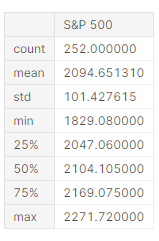


Output

4. Visualisasikan & ringkas nilai harian untuk S&P 500  
Mari kita juga melihat lebih dekat nilai S&P 500, tolok ukur kita.

benchmark\_data.plot()  
plt\_title = ‘S&P 500’  
plt.show()  
benchmark\_data.describe()





Output

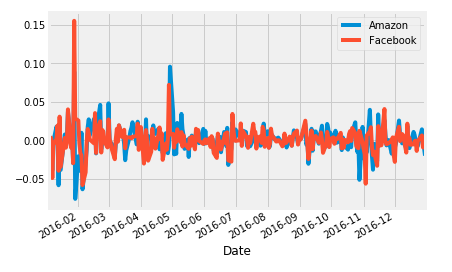
5. Input untuk Sharpe Ratio: Dimulai dengan Pengembalian Saham Harian  
Rasio Sharpe menggunakan perbedaan hasil antara dua peluang investasi yang dipertimbangkan.

Namun, data menunjukkan nilai historis dari setiap investasi, bukan pengembaliannya. Untuk menghitung pengembalian, kita perlu menghitung persentase perubahan nilai dari satu hari ke hari berikutnya. Kami menghitung Rasio Sharpe.

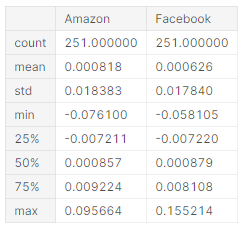
# calculate daily stock\_data returns  
stock\_returns = stock\_data.pct\_change()

# plot the daily returns  
stock\_returns.plot()  
plt.show()

# summarize the daily returns  
stock\_returns.describe()



Output

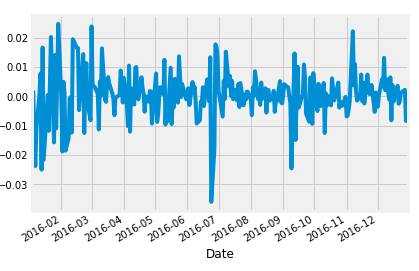


Output

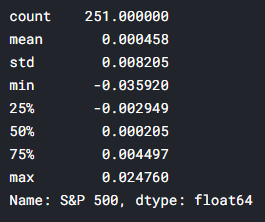
6. Pengembalian S&P 500 harian Untuk S&P 500, menghitung pengembalian harian bekerja dengan cara yang sama, kita hanya perlu memastikan bahwa kita memilihnya sebagai Seri menggunakan tanda kurung tunggal [] dan bukan sebagai DataFrame untuk memfasilitasi penghitungan di langkah berikutnya.

# calculate the mean of excess\_returns  
# … YOUR CODE FOR TASK 8 HERE …  
avg\_excess\_return = excess\_returns.mean()

# plot avg\_excess\_returns  
avg\_excess\_return.plot.bar()  
plt\_title = ‘Mean of the Return Difference’  
plt.show()



Output



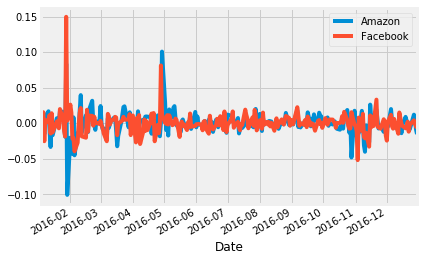
Output

7. Menghitung Kelebihan Pengembalian untuk Amazon dan Facebook vs. S&P 500  
Selanjutnya, kita perlu menghitung kinerja relatif saham vs. tolok ukur S&P 500. Ini dihitung sebagai perbedaan pengembalian antara stock\_returns dan sp\_returns untuk setiap hari.

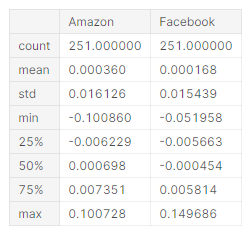
# calculate the difference in daily returns  
excess\_returns = stock\_returns.sub(sp\_returns, axis=0)

# plot the excess\_returns  
excess\_returns.plot()  
plt.show()

# summarize the excess\_returns  
excess\_returns.describe()



Output

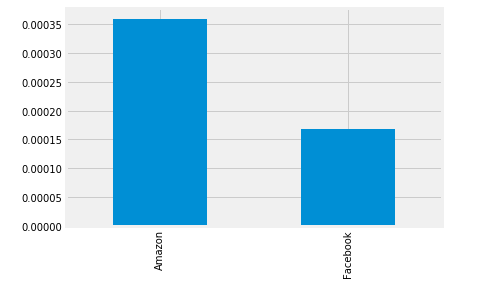


Output

8. Rasio Sharpe, Langkah 1: Perbedaan Rata-Rata dalam Pengembalian Harian Saham vs S&P 500  
Sekarang kita akhirnya dapat mulai menghitung Rasio Sharpe. Pertama kita perlu menghitung rata-rata kelebihan\_kembali. Ini memberi tahu kita berapa banyak lebih atau kurang hasil investasi per hari dibandingkan dengan patokan.

# calculate the mean of excess\_returns  
# … YOUR CODE FOR TASK 8 HERE …  
avg\_excess\_return = excess\_returns.mean()

# plot avg\_excess\_returns  
avg\_excess\_return.plot.bar()  
plt\_title = ‘Mean of the Return Difference’  
plt.show()



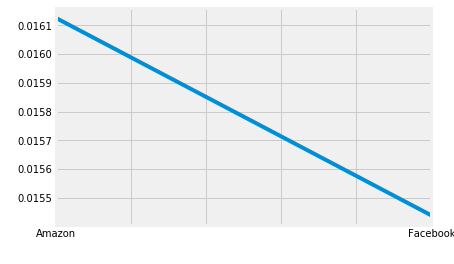
Output

9. Rasio Sharpe, Langkah 2: Deviasi Standar dari Perbedaan Pengembalian  
Sepertinya ada sedikit perbedaan antara pengembalian harian rata-rata untuk Amazon dan Facebook.

Selanjutnya, kita akanmenghitung deviasi standar dari excess\_returns. Ini menunjukkan kepada kita jumlah risiko yang disiratkan oleh investasi dalam saham dibandingkan dengan investasi di S&P 50

# calculate the standard deviations  
sd\_excess\_return = excess\_returns.std()

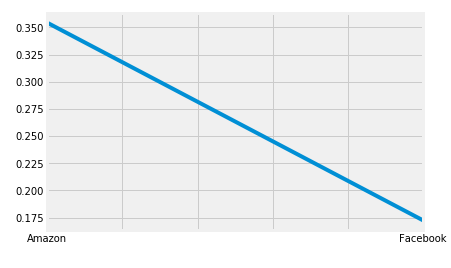
# plot the standard deviations  
sd\_excess\_return.plot()  
plt\_title = ‘Standard Deviation of the Return Difference’  
plt.show()



Output

10. Menyatukan semuanya  
Sekarang kita hanya perlu menghitung rasio avg\_excess\_returns dan sd\_excess\_returns. Hasilnya sekarang adalah rasio Sharpe dan menunjukkan berapa banyak lebih (atau kurang) pengembalian peluang investasi dengan pertimbangan hasil per unit risiko.

Rasio Sharpe sering kali disetahunkan dengan mengalikannya dengan akar kuadrat dari jumlah periode. Kami telah menggunakan data harian sebagai input, jadi kami akan menggunakan akar kuadrat dari jumlah hari perdagangan (5 hari, 52 minggu, dikurangi beberapa hari libur): √252



Output

11. Kesimpulan  
Mengingat dua rasio Sharpe, investasi mana yang harus kita jalani? Pada 2016, Amazon memiliki rasio Sharpe dua kali lebih tinggi dari Facebook. Ini berarti bahwa investasi di Amazon menghasilkan dua kali lebih banyak dibandingkan dengan S&P 500 untuk setiap unit risiko yang akan diasumsikan oleh investor. Dengan kata lain, dalam istilah yang disesuaikan dengan risiko, investasi di Amazon akan lebih menarik.

Perbedaan ini sebagian besar didorong oleh perbedaan hasil, bukan risiko antara Amazon dan Facebook. Risiko memilih Amazon daripada FB (sebagaimana diukur dengan deviasi standar) hanya sedikit lebih tinggi sehingga rasio Sharpe yang lebih tinggi untuk Amazon berakhir lebih tinggi terutama karena pengembalian harian rata-rata yang lebih tinggi untuk Amazon.

Ketika dihadapkan dengan alternatif investasi yang menawarkan pengembalian dan risiko yang berbeda, Rasio Sharpe membantu membuat keputusan dengan menyesuaikan pengembalian dengan perbedaan risiko dan memungkinkan investor untuk membandingkan peluang investasi dengan persyaratan yang sama, yaitu, pada ‘apples-to-apples’ basis.

# Uncomment your choice.  
# buy\_amazon = True  
# buy\_facebook = True

# Proyek DATACAMP: Menggambar Bunga

# 

Bunga

1. Pola di alam

“Ilmuwan tidak mempelajari alam karena berguna; dia mempelajarinya karena dia menyukainya, dan dia menyukainya karena indah.” (Henri Poincaré)Ada banyak contoh fakta alam yang bisa dideskripsikan dalam istilah matematika. Contoh bagusnya adalah bentuk kepingan salju, geometri fraktal romanesco brokoli atau bagaimana kesamaan diri mengatur pertumbuhan tanaman.R adalah alat untuk melakukan analisis serius, tetapi tidak semua hal dalam hidup ini serius. Hidup juga lucu, dan R bisa digunakan untuk bersenang-senang dan melakukan hal-hal indah.

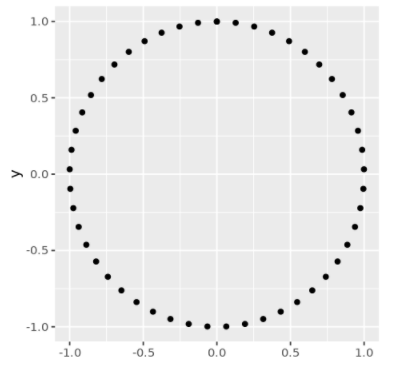
Kekuatan grafisnya dapat digunakan untuk menghasilkan gambar artistik seperti yang mengilustrasikan bagian ini, yang diilhami oleh bagaimana tanaman menata daunnya. Fakta ini disebut phyllotaxis dan akan menjadi dasar dari proyek ini. Di notebook ini, menggunakan paket ggplot2. Selain bersenang-senang, kita akan belajar banyak fitur penting darinya yang berguna tidak hanya untuk seni tetapi juga untuk merepresentasikan data dalam masalah kehidupan nyata. Mari kita mulai dengan memuat perpustakaan.

2. Pemanasan: menggambar titik pada lingkaran  
 Ada banyak cara untuk merepresentasikan data dengan ggplot2: dari plot sebar sederhana hingga plot biola yang lebih kompleks. Fungsi yang dimulai dengan geom\_ menentukan tipe plot. Di notebook ini, kami hanya akan bekerja dengan geom\_point () yang memplot poin dalam dua dimensi. Kami membutuhkan kumpulan data dengan dua variabel; sebut saja mereka x dan y.

Kita akan mulai dengan menggambar 50 titik pada lingkaran berjari-jari 1. Karena setiap titik (x, y) harus berada dalam lingkaran satuan, maka x² + y² = 1. Kita bisa mendapatkan ini menggunakan identitas trigonometri Pythagoras yang sangat terkenal yang menyatakan bahwa sin² (θ) + cos² (θ) = 1 untuk bilangan real apa pun θ.

# Create circle data to plot  
t <- seq(0, 2\*pi, length.out = 50)  
x <- sin(t)  
y <- cos(t)  
df <- data.frame(t, x, y)

# Make a scatter plot of points in a circle  
p <- ggplot(df, aes(x, y))  
p + geom\_point(

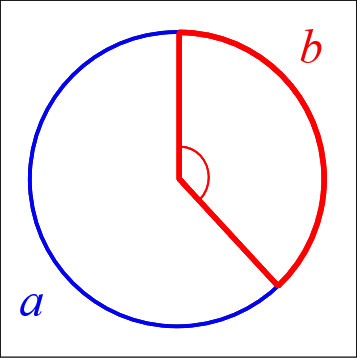


Output

3. Buat selaras dengan Golden Angle  
 Cara sederhana untuk menyusunnya dalam bentuk spiral adalah dengan mengalikan x dan y dengan faktor yang bertambah untuk setiap titik. Kita dapat menggunakan t sebagai faktor tersebut, karena memenuhi kondisi ini, tetapi kami akan melakukan sesuatu yang lebih harmonis. Kita akan menggunakan Golden Angle:

Golden Angle= π (3 — √5)

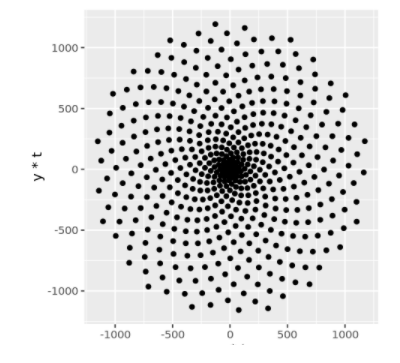
Angka ini terinspirasi dari Golden Ratio, salah satu angka paling terkenal dalam sejarah matematika. Bayangkan Anda memiliki sebuah keliling dan Anda memecahnya menjadi dua busur dengan panjang a dan b, dengan a> b (busur adalah bagian dari keliling). Sudut yang memutus lingkaran sehingga a / b = (a + b) / a disebut Golden Angle. Dengan kata lain: Golden Angle memecah lingkaran sehingga rasio busur besar dan busur kecil adalah Rasio Emas.



Sudut Emas adalah sudut yang ditubuhkan oleh busur yang lebih kecil (merah). Rasio Emas dan Sudut Emas muncul di tempat-tempat yang tidak terduga di alam. Selain kelopak bunga dan daun tanaman, Anda akan menemukannya di kepala biji, kerucut pinus, biji bunga matahari, cangkang, galaksi spiral, badai, dll.

# Define the number of points  
# …. YOUR CODE FOR TASK 3 ….  
points <- 500  
# Define the Golden Angle  
# …. YOUR CODE FOR TASK 3 ….  
angle <-(pi\*(3-sqrt(5)))  
t <- (1:points) \* angle  
x <- sin(t)  
y <-cos(t)  
df <- data.frame(t, x, y)

# Make a scatter plot of points in a spiral  
p <- ggplot(df, aes(x\*t, y\*t))  
p + geom\_point()



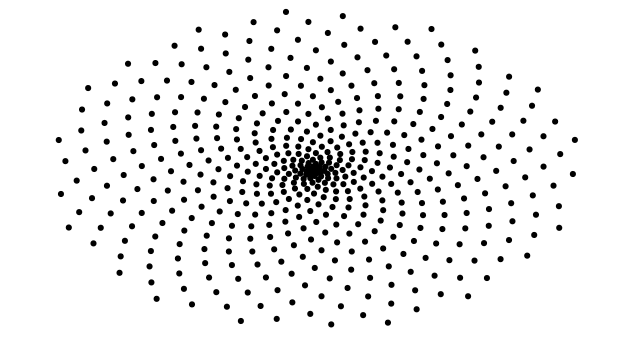
Output

4. Hapus semua yang tidak perlu  
 Selain data, plot mencakup banyak komponen lain yang menentukan tampilan akhirnya. Plot sebelumnya berisi:

latar belakang yang diisi dengan warna abu-abu  
kisi garis putih horizontal dan vertikal  
berdetak di sepanjang sumbu  
judul di setiap sumbu  
teks di sepanjang sumbu ke tanda label  
Seni tidak cocok dengan sebagian besar elemen ini, jadi inilah saatnya untuk bergerak.

df <- data.frame(t, x, y)

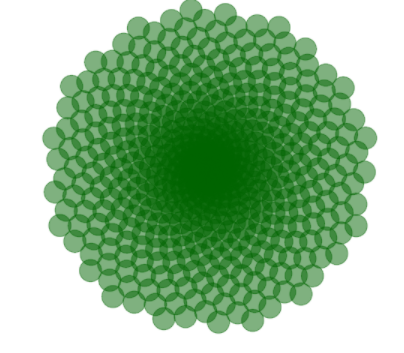
# Make a scatter plot of points in a spiral and remove some plot components  
p <- ggplot(df, aes(x\*t, y\*t))  
p + geom\_point() +  
# …. YOUR CODE FOR TASK 4 ….  
theme(panel.background = element\_rect(fill=”white”),  
panel.grid=element\_blank(),  
axis.text = element\_blank(),  
axis.title = element\_blank(),  
axis.ticks = element\_blank())



Output

5. Sedikit riasan: ukuran, warna dan transparansi  
Gambar akan mulai terlihat seperti tanaman. Dengan mengubah warna, transparansi (disebut juga alpha), dan ukuran titik, gambar akan menjadi lebih menarik.

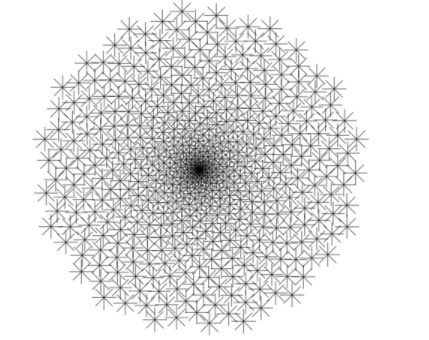
# Change the code from Task 4 to modify the  
# size, transparency, and color of the points  
p <- ggplot(df, aes(x\*t, y\*t))  
p + geom\_point(size = 8, alpha = 0.5, color = “darkgreen”) +  
theme(panel.background = element\_rect(fill=”white”),  
panel.grid=element\_blank(),  
axis.text = element\_blank(),  
axis.title = element\_blank(),  
axis.ticks = element\_blank())



Output

6. Bermainlah dengan estetika: dandelion  
Hingga saat ini, semua titik memiliki tampilan yang sama (ukuran, warna, bentuk, dan alpha). Terkadang kita ingin membuat tampilan poin bergantung pada variabel dalam dataset. Sekarang kita akan membuat variabel ukuran. Kami juga akan mengubah bentuk poin. Meskipun kami tidak dapat meniupnya, gambar yang dihasilkan harus mengingatkan Anda pada dandelion.

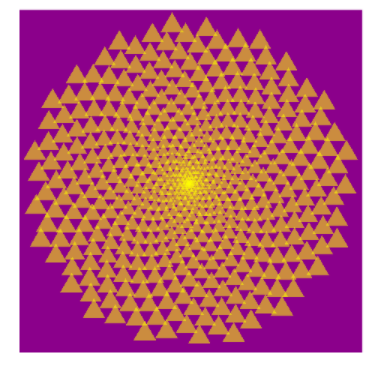
# Copy the code from Task 5 and modify the  
# color, size, and shape of the points  
p <- ggplot(df, aes(x\*t, y\*t))  
p + geom\_point(aes(size = t, alpha = 0.5), shape = 8) +  
theme(panel.background = element\_rect(fill=”white”),  
panel.grid=element\_blank(),  
axis.text = element\_blank(),  
axis.title = element\_blank(),  
axis.ticks = element\_blank(),  
legend.position=”none”)



Output

7. Gabungkan semuanya: bunga matahari  
Tanaman tidak hanya menggunakan Golden Angle untuk menyusun daun. Golden Angle juga ditemukan pada susunan biji bunga matahari. kita hanya perlu menggabungkan beberapa hal yang sudah kita ketahui.

# Copy the code from Task 6 and modify the color and  
# shape of the points, and the background color  
p <- ggplot(df, aes(x\*t, y\*t))  
p + geom\_point(aes(size = t, alpha = 0.5), shape = 17, color = “yellow”) +  
theme(panel.background = element\_rect(fill=”darkmagenta”),  
panel.grid=element\_blank(),  
axis.text = element\_blank(),  
axis.title = element\_blank(),  
axis.ticks = element\_blank(),  
legend.position=”none”)



Output

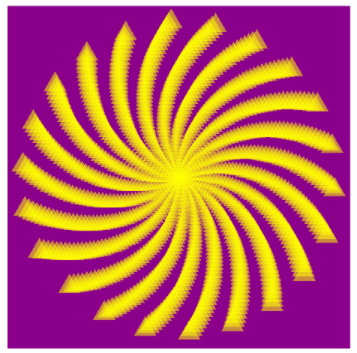
8. Bagaimana jika Anda mengubah sudutnya?  
Pola ini sangat sensitif terhadap sudut antar titik yang membentuk spiral. Perubahan kecil pada sudut dapat menghasilkan gambar yang sangat berbeda. Mari kita lihat contohnya.

# Change the value of the angle  
angle <- 2  
points <- 1000

t <- (1:points)\*angle  
x <- sin(t)  
y <- cos(t)  
df <- data.frame(t, x, y)

# Copy the plotting code from Task 7  
# …. YOUR CODE FOR TASK 8 ….

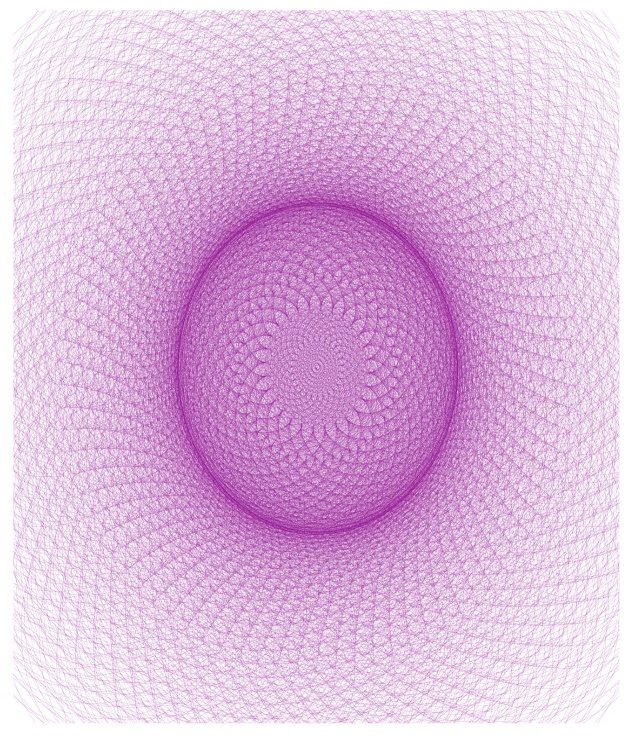
p <- ggplot(df, aes(x\*t, y\*t))  
p + geom\_point(aes(size = t, alpha = 0.5), shape = 17, color = “yellow”) +  
theme(panel.background = element\_rect(fill=”darkmagenta”),  
panel.grid=element\_blank(),  
axis.text = element\_blank(),  
axis.title = element\_blank(),  
axis.ticks = element\_blank(),  
legend.position=”none”)



Output

9. Semua bersama sekarang: bunga imajiner

Teknik yang telah di gunakan sejauh ini memungkinkan untuk membuat pola dalam jumlah tak terbatas yang terinspirasi oleh alam: satu-satunya batasan adalah imajinasi Tetapi membuat karya seni juga menjadi alasan yang menyenangkan untuk belajar menggunakan ggplot2. Semua trik yang telah kita lihat



Gambar di atas adalah variasi sederhana dari bunga sebelumnya dan pada dasarnya sangat mirip dengan gambar pertama di mana kami memplot 50 titik dalam sebuah lingkaran.

# Proyek DataCamp: Dr. Semmelweis dan Penemuan Cuci Tangan

# 

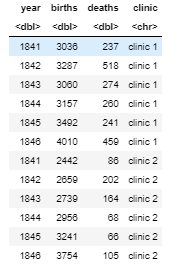
Dr. Ignaz Semmelweis

Ini adalah Dr. Ignaz Semmelweis, seorang dokter Hongaria yang lahir pada tahun 1818 dan aktif di Rumah Sakit Umum Wina. Jika Dr. Semmelweis terlihat bermasalah, itu mungkin karena dia memikirkan demam nifas: Penyakit mematikan yang menyerang wanita yang baru saja melahirkan. Dia memikirkannya karena pada awal tahun 1840-an di Rumah Sakit Umum Wina sebanyak 10% wanita yang melahirkan meninggal karenanya. Dia memikirkannya karena dia tahu penyebab demam nifas: Ini adalah tangan yang terkontaminasi dari para dokter yang melahirkan bayi. Dan mereka tidak mau mendengarkan dia dan mencuci tangan mereka!Semmelweis menemukan pentingnya mencuci tangan. Dimulai dengan melihat data yang membuat Semmelweis menyadari bahwa ada yang salah dengan prosedur di Rumah Sakit Umum Wina.

# Load in the tidyverse package  
# …. YOUR CODE FOR TASK 1 ….  
library(tidyverse)

# Read datasets/yearly\_deaths\_by\_clinic.csv into yearly  
yearly <- read\_csv(“datasets/yearly\_deaths\_by\_clinic.csv”)

# Print out yearly  
# …. YOUR CODE FOR TASK 1 ….  
yearly

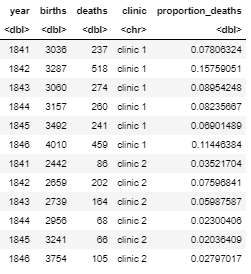


Output yearly

2. Jumlah kematian yang mengkhawatirkan  
Tabel di atas menunjukkan jumlah wanita yang melahirkan di dua klinik di Rumah Sakit Umum Wina dari tahun 1841 hingga 1846. Anda akan melihat bahwa melahirkan sangat berbahaya; sejumlah besar perempuan meninggal akibat melahirkan, kebanyakan dari mereka karena demam nifas.Hal ini terlihat lebih jelas jika kita melihat proporsi kematian dari jumlah perempuan yang melahirkan.

# Adding a new column to yearly with proportion of deaths per no. births  
# …. YOUR CODE FOR TASK 1 ….  
yearly <- yearly %>% mutate(proportion\_deaths = deaths/births)

# Print out yearly  
yearly

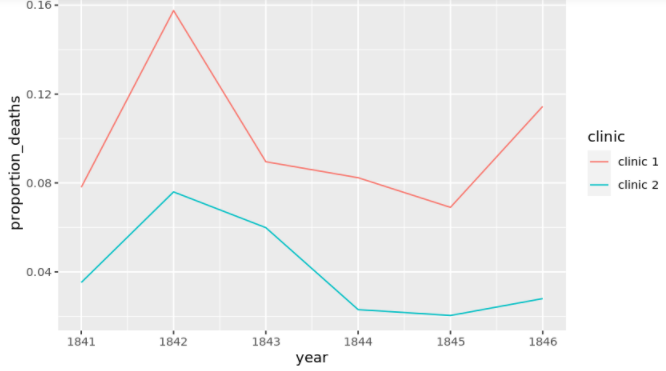


Proporsi kematian dari melahirkan

3. Kematian di klinik  
Jika sekarang diplot proporsi kematian di klinik 1 dan klinik 2, maka akan terlihat pola yang aneh…

# Setting the size of plots in this notebook  
options(repr.plot.width=7, repr.plot.height=4)

# Plot yearly proportion of deaths at the two clinics  
# …. YOUR CODE FOR TASK 3 ….  
ggplot(yearly, aes(x=year, y = proportion\_deaths, col = clinic)) + geom\_line()

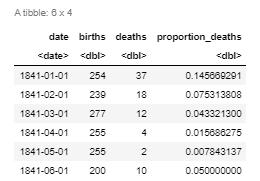


Output

4. Cuci tangan dimulai  
Mengapa proporsi kematian terus-menerus jauh lebih tinggi di Klinik 1? Semmelweis melihat pola yang sama dan bingung serta tertekan. Satu-satunya perbedaan antara kedua klinik adalah banyaknya mahasiswa kedokteran yang bertugas di Klinik 1, sedangkan sebagian besar mahasiswa bidan bertugas di Klinik 2. Sementara bidan hanya merawat ibu yang melahirkan, mahasiswa kedokteran juga menghabiskan waktu di ruang otopsi untuk memeriksa jenazah.Semmelweis mulai curiga ada sesuatu pada mayat yang menyebar dari tangan mahasiswa kedokteran itu yang menyebabkan demam nifas. Jadi dalam upaya putus asa untuk menghentikan angka kematian yang tinggi, dia memutuskan: Cuci tanganmu! Ini adalah permintaan yang tidak ortodoks dan kontroversial, tidak ada orang di Wina yang tahu tentang bakteri pada saat ini.lalu saya memulai mengolah data dari mencuci tangan untuk membuktikan apakah mencuci tangan memiliki efek.

# Read datasets/monthly\_deaths.csv into monthly  
monthly <- read\_csv(“datasets/monthly\_deaths.csv”)

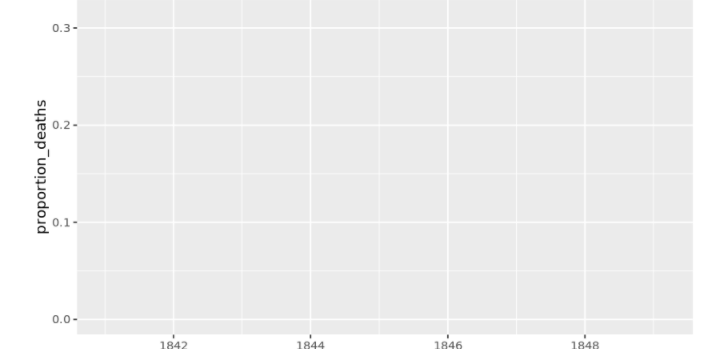
# Adding a new column with proportion of deaths per no. births  
# …. YOUR CODE FOR TASK 4 ….  
monthly <- monthly %>% mutate(proportion\_deaths=deaths/births)  
# Print out the first rows in monthly  
# …. YOUR CODE FOR TASK 4 ….  
head(monthly)



Output

5. Efek mencuci tangan  
Dengan data yang dimuat, sekarang dapat dilihat proporsi kematian dari waktu ke waktu. Dalam plot di bawah ini belum ditandai di mana mencuci tangan wajib dimulai, tetapi itu mengurangi proporsi kematian sedemikian rupa sehingga Anda dapat melihatnya!

# Plot monthly proportion of deaths  
# … YOUR CODE FOR TASK 5 …  
ggplot(monthly, aes(x = date, y = proportion\_deaths))



Output

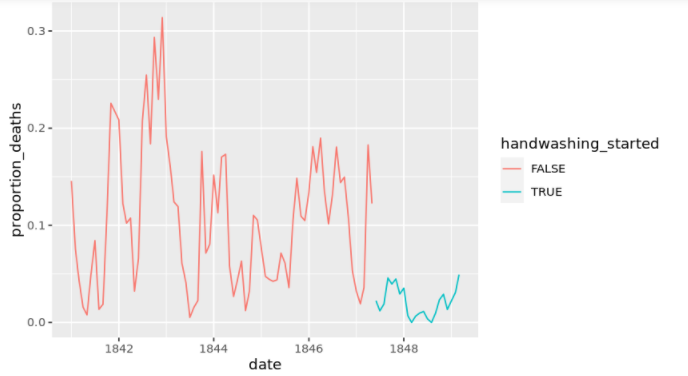
6. Efek mencuci tangan disorot  
Mulai musim panas 1847 proporsi kematian berkurang drastis dan, ya, saat itulah Semmelweis mewajibkan cuci tangan. Efek mencuci tangan menjadi lebih jelas jika disorot dalam grafik.

# From this date handwashing was made mandatory  
handwashing\_start = as.Date(‘1847–06–01’)

# Add a TRUE/FALSE column to monthly called handwashing\_started  
# …. YOUR CODE FOR TASK 6 ….  
monthly <- monthly %>%

# Plot monthly proportion of deaths before and after handwashing  
# …. YOUR CODE FOR TASK 6 ….  
mutate(handwashing\_started = ifelse(date >= handwashing\_start, TRUE, FALSE))

# Plot monthly proportion of deaths before and after handwashing  
ggplot(monthly, aes(x = date, y = proportion\_deaths, col = handwashing\_started)) +  
geom\_line()



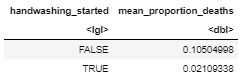
Output

7. Lebih banyak mencuci tangan, lebih sedikit kematian?  
Sekali lagi, grafik tersebut menunjukkan bahwa mencuci tangan memiliki pengaruh yang sangat besar. Berapa rata-rata hal itu mengurangi proporsi kematian bulanan?

# Calculating the mean proportion of deaths  
# before and after handwashing.

monthly\_summary <- monthly %>% group\_by(handwashing\_started) %>%  
summarise(mean\_proportion\_deaths = mean(proportion\_deaths))  
# …. YOUR CODE FOR TASK 7 HERE ….

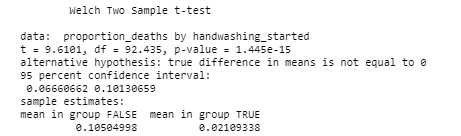
# Printing out the summary.  
monthly\_summar



Output

8. Analisis statistik data cuci tangan Semmelweis  
Ini mengurangi proporsi kematian sekitar 8 poin persentase! Dari rata-rata 10% sebelum mencuci tangan menjadi hanya 2% ketika mencuci tangan diberlakukan (yang masih merupakan angka yang tinggi menurut standar modern). Untuk merasakan ketidakpastian seputar seberapa banyak mencuci tangan mengurangi kematian, dapat digunakan penggunaaninterval kepercayaan (di sini dihitung menggunakan uji-t).

# Calculating a 95% Confidence intrerval using t.test  
test\_result <- t.test( proportion\_deaths ~ handwashing\_started, data = monthly)  
test\_result



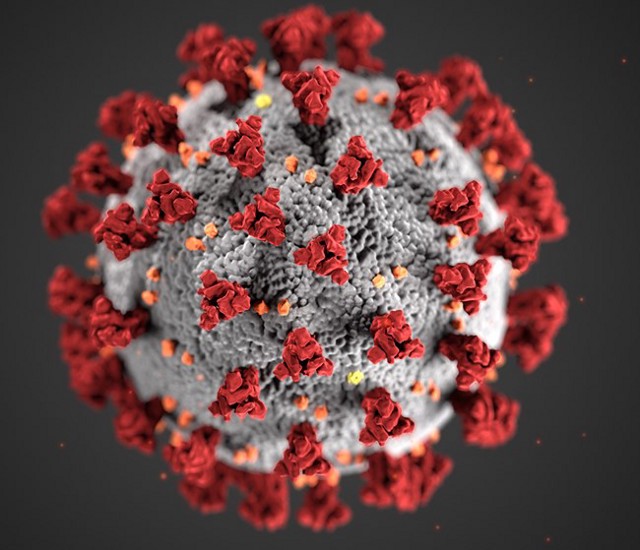
Output

9. Nasib Dr. Semmelweis  
Fakta bahwa para dokter tidak mencuci tangan meningkatkan proporsi kematian antara 6,7 ​​dan 10 poin persentase, menurut interval kepercayaan 95%. Secara keseluruhan, tampaknya Semmelweis memiliki bukti kuat bahwa cuci tangan adalah prosedur sederhana namun sangat efektif yang dapat menyelamatkan banyak nyawa. Tragisnya adalah, terlepas dari bukti, teori Semmelweis — bahwa demam nifas disebabkan oleh beberapa “zat” (yang sekarang kita kenal sebagai bakteri) dari mayat ruang otopsi — diejek oleh para ilmuwan kontemporer. Komunitas medis sebagian besar menolak penemuannya dan pada tahun 1849 dia terpaksa meninggalkan Rumah Sakit Umum Wina untuk selamanya. Salah satu alasannya adalah bahwa statistik dan argumen statistik tidak umum dalam ilmu kedokteran pada tahun 1800-an. Semmelweis hanya mempublikasikan datanya sebagai tabel panjang data mentah, tetapi dia tidak menampilkan grafik atau interval kepercayaan. Jika dia memiliki akses ke analisis yang baru saja kita kumpulkan, dia mungkin lebih berhasil membuat para dokter Wina mencuci tangan.

# The data Semmelweis collected points to that:  
doctors\_should\_wash\_their\_hands <- TRUE

### Visualisasi COVID19

1. Dari epidemi menjadi pandemi



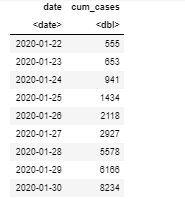
Pada Desember 2019, virus korona COVID-19 pertama kali diidentifikasi di wilayah Wuhan, China. Pada 11 Maret 2020, Organisasi Kesehatan Dunia (WHO) mengkategorikan wabah COVID-19 sebagai pandemi. Banyak yang telah terjadi di bulan-bulan antara wabah besar di Iran, Korea Selatan, dan Italia.Kita tahu bahwa COVID-19 menyebar melalui tetesan pernafasan, seperti melalui batuk, bersin, atau berbicara. Tapi, seberapa cepat virus itu menyebar ke seluruh dunia? Dan, dapatkah kita melihat efek dari kebijakan di seluruh negeri, seperti penghentian dan karantina?

Untungnya, organisasi di seluruh dunia telah mengumpulkan data sehingga pemerintah dapat memantau dan belajar dari pandemi ini. Khususnya, Pusat Sains dan Teknik Sistem Universitas Johns Hopkins membuat penyimpanan data yang tersedia untuk umum untuk mengonsolidasi data ini dari sumber-sumber seperti WHO, Pusat Pengendalian dan Pencegahan Penyakit (CDC), dan Kementerian Kesehatan dari berbagai negara.Harap diperhatikan bahwa informasi dan data terkait COVID-19 sering diperbarui. Data yang digunakan dalam proyek ini ditarik pada 17 Maret 2020, dan tidak boleh dianggap sebagai data terbaru yang tersedia.

# Load the readr, ggplot2, and dplyr packages  
library(readr)  
library(ggplot2)  
library(dplyr)

# Read datasets/confirmed\_cases\_worldwide.csv into confirmed\_cases\_worldwide  
confirmed\_cases\_worldwide <- read\_csv(“datasets/confirmed\_cases\_worldwide.csv”)

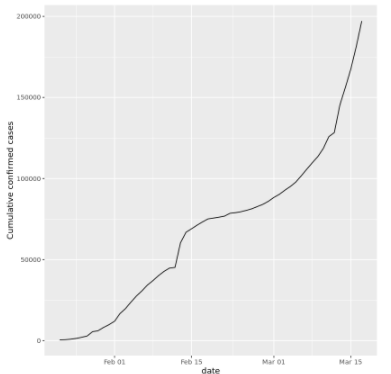
# Print out confirmed\_cases\_worldwide  
confirmed\_cases\_worldwide



Potongan Output

2. Kasus yang dikonfirmasi di seluruh dunia  
Tabel di atas menunjukkan kasus terkonfirmasi kumulatif COVID-19 di seluruh dunia berdasarkan tanggal. Hanya dengan membaca angka dalam tabel, sulit untuk memahami skala dan pertumbuhan wabah. Mari menggambar plot garis untuk memvisualisasikan kasus yang dikonfirmasi di seluruh dunia.

# Draw a line plot of cumulative cases vs. date  
# Label the y-axis  
ggplot(confirmed\_cases\_worldwide, aes(date, cum\_cases)) +  
 geom\_line() +  
 ylab(“Cumulative confirmed cases”)



Output

3. China dibandingkan dengan negara lain di dunia

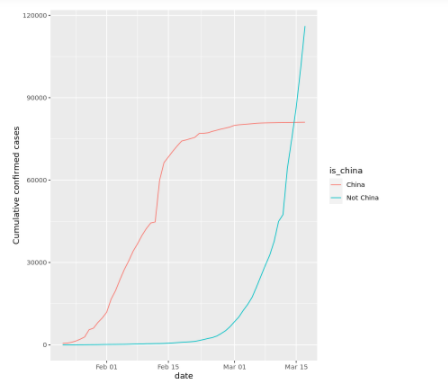
Di awal wabah, kasus COVID-19 terutama berpusat di China. Mari kita plot kasus COVID-19 yang dikonfirmasi di China dan seluruh dunia secara terpisah untuk melihat apakah itu memberi kita wawasan. Kita akan membangun plot ini di tugas-tugas mendatang. Satu hal yang penting untuk tugas berikut ini adalah Anda menambahkan estetika dalam geometri garis ggplot Anda, daripada menjadikannya estetika global.

# Read in datasets/confirmed\_cases\_china\_vs\_world.csv  
confirmed\_cases\_china\_vs\_world <- read\_csv(“datasets/confirmed\_cases\_china\_vs\_world.csv”)

# See the result  
glimpse(confirmed\_cases\_china\_vs\_world)

# Draw a line plot of cumulative cases vs. date, colored by is\_china  
# Define aesthetics within the line geom  
plt\_cum\_confirmed\_cases\_china\_vs\_world <- ggplot(confirmed\_cases\_china\_vs\_world) +  
 geom\_line(aes(date, cum\_cases, color = is\_china)) +  
 ylab(“Cumulative confirmed cases”)

# See the plot  
plt\_cum\_confirmed\_cases\_china\_vs\_world

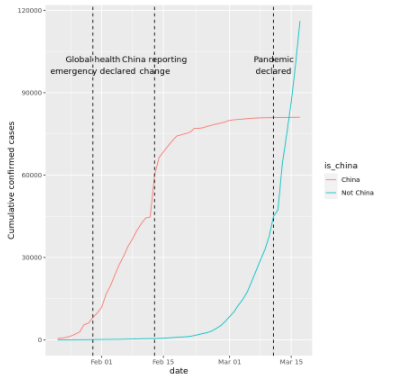


Output

4. Mari beri keterangan!  
 Wow! Kedua garis tersebut memiliki bentuk yang sangat berbeda. Pada bulan Februari, sebagian besar kasus terjadi di China. Itu berubah pada bulan Maret ketika benar-benar menjadi wabah global: sekitar 14 Maret, jumlah total kasus di luar China melampaui kasus di China. Ini terjadi beberapa hari setelah WHO menyatakan pandemi.Ada beberapa peristiwa penting lainnya yang terjadi selama wabah. Misalnya, lonjakan besar di garis China pada 13 Februari 2020 bukan hanya hari yang buruk terkait wabah; China mengubah cara melaporkan angka pada hari itu (CT scan diterima sebagai bukti COVID-19, bukan hanya tes laboratorium).Dengan memberi anotasi peristiwa seperti ini, kami dapat menafsirkan perubahan dalam plot dengan lebih baik.

who\_events <- tribble(  
 ~ date, ~ event,  
 “2020–01–30”, “Global health\nemergency declared”,  
 “2020–03–11”, “Pandemic\ndeclared”,  
 “2020–02–13”, “China reporting\nchange”  
) %>%  
 mutate(date = as.Date(date))

# Using who\_events, add vertical dashed lines with an xintercept at date  
# and text at date, labeled by event, and at 100000 on the y-axis  
plt\_cum\_confirmed\_cases\_china\_vs\_world +  
 geom\_vline(aes(xintercept = date), data = who\_events, linetype = “dashed”) +  
 geom\_text(aes(date, label = event), data = who\_events, y = 1e5)

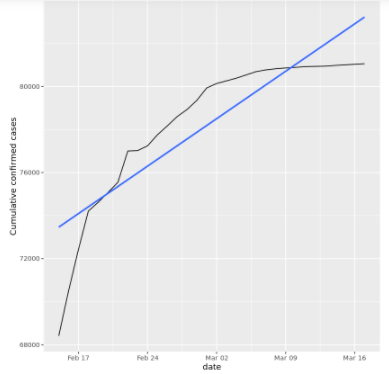


Output

5. Menambahkan garis tren ke China  
 Ketika mencoba menilai seberapa besar masalah yang akan terjadi di masa depan, kami memerlukan ukuran seberapa cepat jumlah kasus berkembang. Titik awal yang baik adalah melihat apakah kasing tumbuh lebih cepat atau lebih lambat daripada secara linier.Ada lonjakan kasus yang jelas sekitar 13 Februari 2020, dengan perubahan pelaporan di China. Namun, beberapa hari kemudian, pertumbuhan kasus di China melambat. Bagaimana kita bisa menggambarkan pertumbuhan COVID-19 di China setelah 15 Februari 2020?

# Filter for China, from Feb 15  
china\_after\_feb15 <- confirmed\_cases\_china\_vs\_world %>%  
 filter(is\_china == “China”, date >= “2020–02–15”)

# Using china\_after\_feb15, draw a line plot cum\_cases vs. date  
# Add a smooth trend line using linear regression, no error bars  
ggplot(china\_after\_feb15, aes(date, cum\_cases)) +  
 geom\_line() +  
 geom\_smooth(method = “lm”, se = FALSE) +  
 ylab(“Cumulative confirmed cases”)



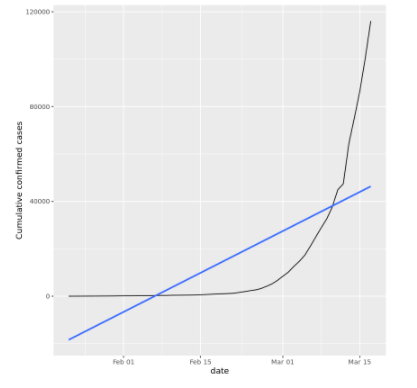
Output

6. Dan bagian dunia lainnya?  
 Dari plot di atas, laju pertumbuhan di China lebih lambat dari linier. Itu berita bagus karena ini menunjukkan China setidaknya telah menahan virus pada akhir Februari dan awal Maret. Bagaimana bagian dunia lainnya dibandingkan dengan pertumbuhan linier?

# Filter confirmed\_cases\_china\_vs\_world for not China  
not\_china <- confirmed\_cases\_china\_vs\_world %>%  
 filter(is\_china == “Not China”)

# Using not\_china, draw a line plot cum\_cases vs. date  
# Add a smooth trend line using linear regression, no error bars  
plt\_not\_china\_trend\_lin <- ggplot(not\_china, aes(date, cum\_cases)) +  
 geom\_line() +  
 geom\_smooth(method = “lm”, se = FALSE) +  
 ylab(“Cumulative confirmed cases”)

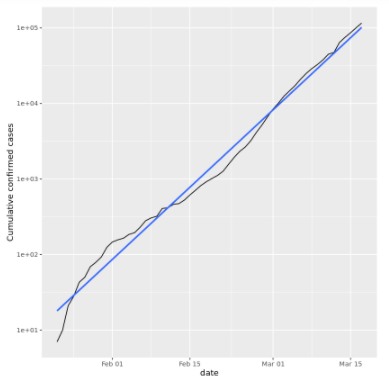
# See the result  
plt\_not\_china\_trend\_lin



Output

7. Menambahkan skala logaritmik  
 Dari plot di atas, kita dapat melihat garis lurus tidak pas sama sekali, dan bagian dunia lainnya tumbuh jauh lebih cepat daripada secara linier. Bagaimana jika kita menambahkan skala logaritmik ke sumbu y?

# Modify the plot to use a logarithmic scale on the y-axis  
plt\_not\_china\_trend\_lin +   
 scale\_y\_log10()



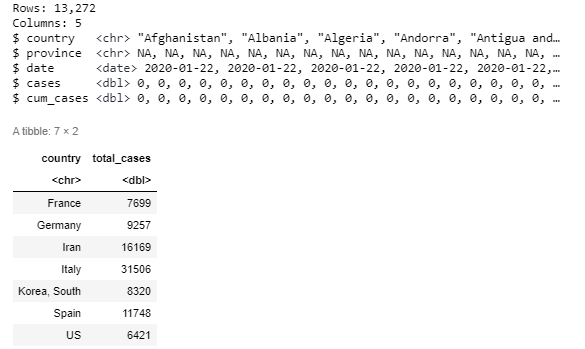
Output

8. Negara mana di luar China yang paling terpukul?   
 Dengan skala logaritmik, kita menjadi lebih cocok dengan data. Dari sudut pandang ilmu data, kecocokan adalah kabar baik. Sayangnya, dari sudut pandang kesehatan masyarakat, itu berarti kasus COVID-19 di seluruh dunia tumbuh dengan kecepatan eksponensial, yang merupakan berita buruk.Tidak semua negara dipengaruhi oleh COVID-19 secara merata, dan akan sangat membantu untuk mengetahui di belahan dunia mana masalahnya paling besar. Mari temukan negara di luar China dengan kasus terkonfirmasi paling banyak di kumpulan data ini.

# Run this to get the data for each country  
confirmed\_cases\_by\_country <- read\_csv(“datasets/confirmed\_cases\_by\_country.csv”)  
glimpse(confirmed\_cases\_by\_country)

# Group by country, summarize to calculate total cases, find the top 7  
top\_countries\_by\_total\_cases <- confirmed\_cases\_by\_country %>%  
 group\_by(country) %>%  
 summarize(total\_cases = max(cum\_cases)) %>%  
 top\_n(7, total\_cases)

# See the result  
top\_countries\_by\_total\_cases

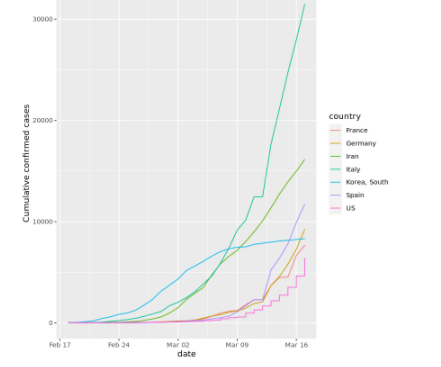
output

9. Negara-negara yang terkena dampak paling parah pada Pertengahan Maret 2020  
 Meskipun wabah pertama kali diidentifikasi di China, hanya ada satu negara dari Asia Timur (Korea Selatan) pada tabel di atas. Empat dari negara yang terdaftar (Prancis, Jerman, Italia, dan Spanyol) berada di Eropa dan berbatasan langsung. Untuk mendapatkan lebih banyak konteks, kami dapat memplot kasus terkonfirmasi negara-negara ini dari waktu ke waktu.

# Read in the dataset from datasets/confirmed\_cases\_top7\_outside\_china.csv  
confirmed\_cases\_top7\_outside\_china <- read\_csv(“datasets/confirmed\_cases\_top7\_outside\_china.csv”)

# Glimpse at the contents of confirmed\_cases\_top7\_outside\_china  
glimpse(confirmed\_cases\_top7\_outside\_china)

# Using confirmed\_cases\_top7\_outside\_china, draw a line plot of  
# cum\_cases vs. date, colored by country  
ggplot(confirmed\_cases\_top7\_outside\_china, aes(date, cum\_cases, color = country)) +  
 geom\_line() +  
 ylab(“Cumulative confirmed cases”)



Output