DataScientist Society

データサイエンス100本ノック

構造化データ加工編

2020.06.15

データサイエンス100本ノック（構造化データ加工編）の狙い

データ活用の重要性に対する認知が広がる中で、データサイエンティストの必要性も増々高まっています。そのような中で、書籍やWebサイトなど、自己研鑽に必要な情報源も多く提供されています。しかし、実践するための「データ」や「プログラミング実行環境」を持ち合わせていないため、「実践力」を身につけることが難しい、というケースが見られます。

そこでデータサイエンティスト協会では、データサイエンス100本ノック（構造化データ加工編）をリリースいたしました。データと実行環境構築スクリプト、そして演習問題をワンセットにして公開しています。

自然言語処理や画像処理、深層学習などの実践練習環境は、すでに素晴らしい題材が公開されています。例えば深層学習については、東京大学 松尾研究室「Deep Learning基礎講座演習コンテンツ 公開ページ」にてnotebookファイルが公開されており、Google Colaboratoryなどを利用することで無料で実践可能となります。

一方、構造化データについては、無料で利用できるオープンデータが多数あるものの、データ分析力の実践という観点で環境整備がなされたものはあまり多くありません。分析実務においては構造化データの活用が多くを占めるという実態を鑑み、「構造化データ加工編」というサブタイトルで公開させていただきました。

データの加工・集計、統計学や機械学習を駆使したモデリングの前処理など、基礎的なデータハンドリングの修行場として利用いただければと思います。

演習問題の構成

- 実行環境のサポート言語はSQL、Python、Rとなります
- それぞれの言語で、同様の設問を100問用意しています
- 各設問に対する解答例のファイルも用意しています

注）設問によっては、向かない言語もありますが、実際の実務では企業におけるセキュリティやシステム実装の観点などから、どうしてもSQLでやらねばならないケース、Pythonでやらねばならないケース、Rでやらねばならないケースなどが出てきます。そのため、リソースを過剰に使ってしまうなどあまり良いコード例ではないものもありますが、分析で良く利用される3言語について、「この言語のときはこうすればできる」という横軸を揃えた構成することを優先しています。

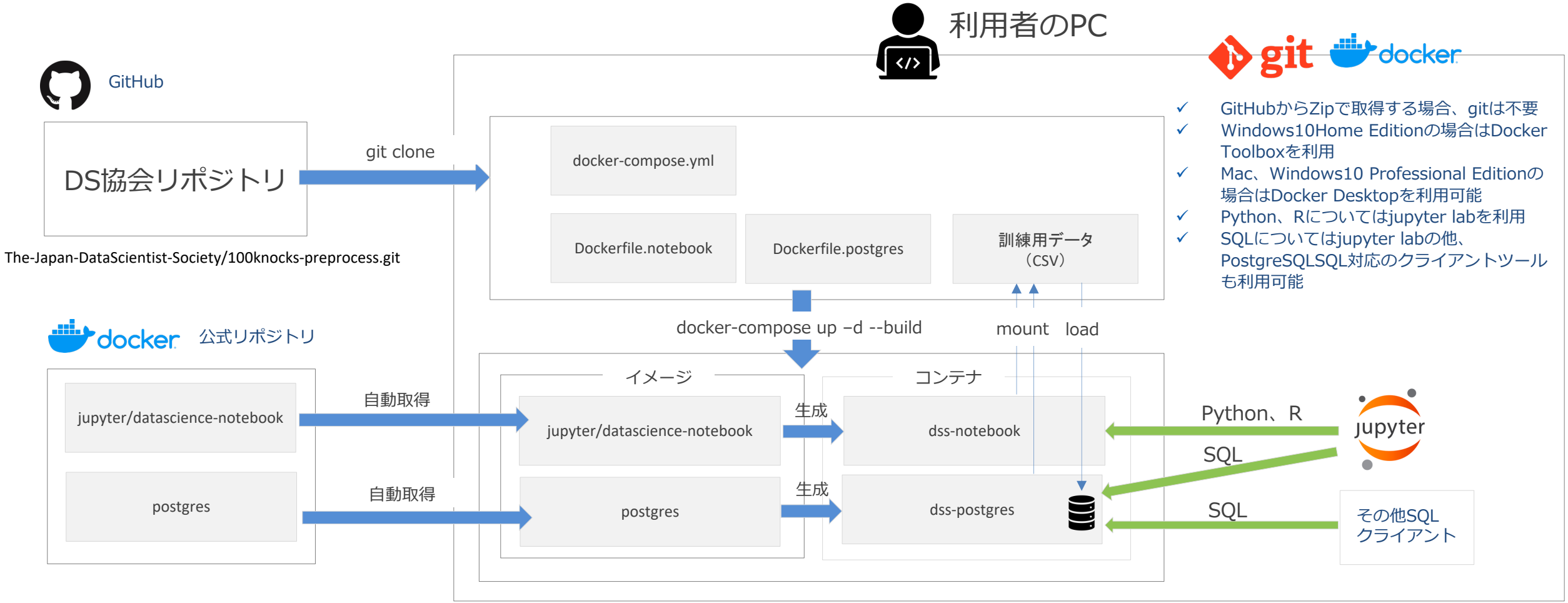
No.	大区分	設問数
1	列に対する操作	3
2	行に対する操作	6
3	あいまい条件	7
4	ソート	4
5	集計	13
6	副問合せ	2
7	結合	7
8	縦横変換	2
9	データ変換	14
10	数値変換	4

No.	大区分	設問数
11	四則演算	7
12	日付型の計算	5
13	サンプリング	2
14	外れ値・異常値	2
15	欠損値	5
16	除算エラー対応	1
17	座標データ	2
18	名寄せ	2
19	データ分割	2
20	不均衡データ	1

No.	大区分	設問数
21	正規化・非正規化	2
22	ファイル入出力	7
合計		100

実践環境

100本ノック（構造化データ加工編）の仕組みは以下の通りとなります。



必要なPCリソースおよびソフトウェア

OS	✓ macOS 10.13 以上 ✓ Windows10 Professional Edition ✓ Windows10 Home Edition
メモリ	✓ 8GB以上（Dockerへの割当4GB以上）を推奨
HDD	✓ 15GB以上の空きスペース
ソフトウェア	✓ Docker DesktopまたはDocker Toolbox ✓ git（GitHubからZIPで取得する場合は不要）

Dockerの概要とファイル共有

- Dockerは、Linuxのコンテナ技術を使った仮想環境であり、PCの中であたかも別のLinux OSマシンがあるかのように動かすことができるソフトウェアです
- Dockerの中のLinuxをゲストOSと呼び、PC本体のOSをホストOSと呼びます
- OSおよび様々なソフトウェアがインストールされたDockerイメージとよばれるファイルが公開されており、DockerイメージからDockerコンテナと呼ばれるプロセスを立ち上げることでPC内に仮想環境を実現します
- データサイエンス100本ノック（構造化データ加工編）では、以下２つのコンテナが起動するよう設定されています（図1）
  - PythonおよびRの実行環境であるdss-notebook（ポート番号：8888）
  - PostgreSQLによるSQL実行環境であるdss-postgres（port番号：5432）
- 複数のコンテナを管理する際に便利なdocker-composeという機能を利用しています
- ゲストOSからホストOSのディレクトリをマウントすることが可能であり、これによりホストOSとゲストOSとのファイル受け渡しを容易に行うことができます
- それぞれのコンテナから、データサイエンス100本ノック（構造化データ加工編）のディレクトリが自動的にマウントされるよう設定されています(図2)
- PCと仮想環境を隔離し、分析環境構築に伴うライブラリインストールなどでマシン本体を汚してしまうことがないため、実際の分析実務でもよく利用されます

図1：Dockerの構成イメージ

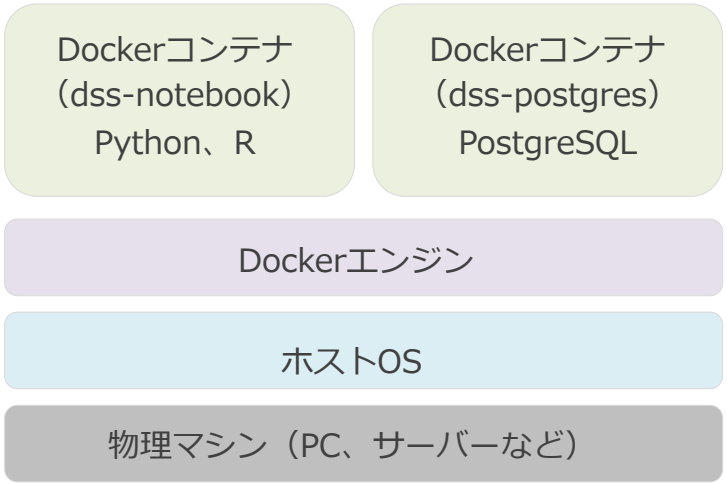
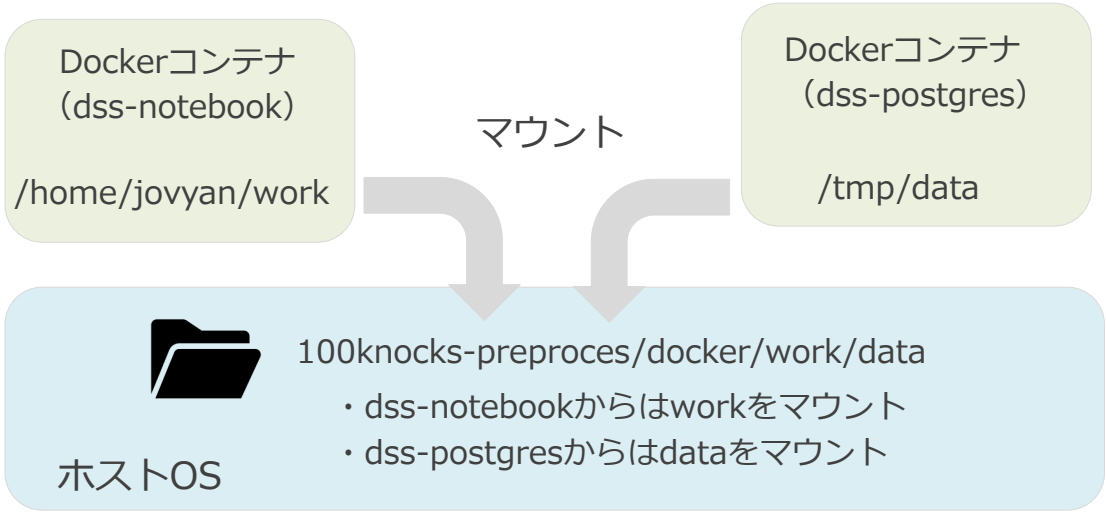


図2：コンテナによるディレクトリ共有



環境構築手順

以下の手順で環境を構築します。なお、③の手順ではgitの機能を使わずzipファイルで100本ノック関連ファイルを取得することも可能です。その場合は、①の手順が不要となります。

① gitをインストールする（詳細は関連書籍やWebサイト等を参照願います）

② Dockerをインストールする※1（詳細は関連書籍やWebサイト等を参照願います）

③ 100本ノックリポジトリをクローンする※２（git clone git@github.com:The-Japan-DataScientist-Society/100knocks-preprocess.git）

④ ターミナル等※3でdocker-composeファイルのあるディレクトリまで移動（cd 100knocks-preprocess）

⑤ ターミナル等※3でコンテナ作成コマンドを実行する（docker-compose up -d --build）

※1: Windows10 Home Editionの場合はDocker Toolboxを利用してください。Mac OSXとWindows 10 Professional EditionはDocker DesktopとDocker Toolboxのどちらも利用可能ですが、情報源の豊富さからDocker Desktopを推奨いたします。

※2: 自身のユーザーホームディレクトリ配下に取得すると設定作業がもっとも簡単になります。それ以外のディレクトリに格納する場合、Dockerの共有ディレクトリ設定が必要になります（Docker Desktop設定のページ、またはDocker Toolbox設定のページを参照願います）。

※3、Docker Desktopの場合はターミナル（OSX）またはコマンドプロンプト（Windows）、Docker Toolboxの場合はDocker Quickstart Terminalを使用します。

ダウンロード時間を除けば作業時間は10分程度となります。

docker-compose 基本コマンド

docker-composeという機能を使うことで、複数のコンテナ管理を行うことができます。docker-compose.ymlファイルが存在するディレクトリで実行するか、当ファイルをオプションで指定する必要があります。以下に基本コマンドを記載します。

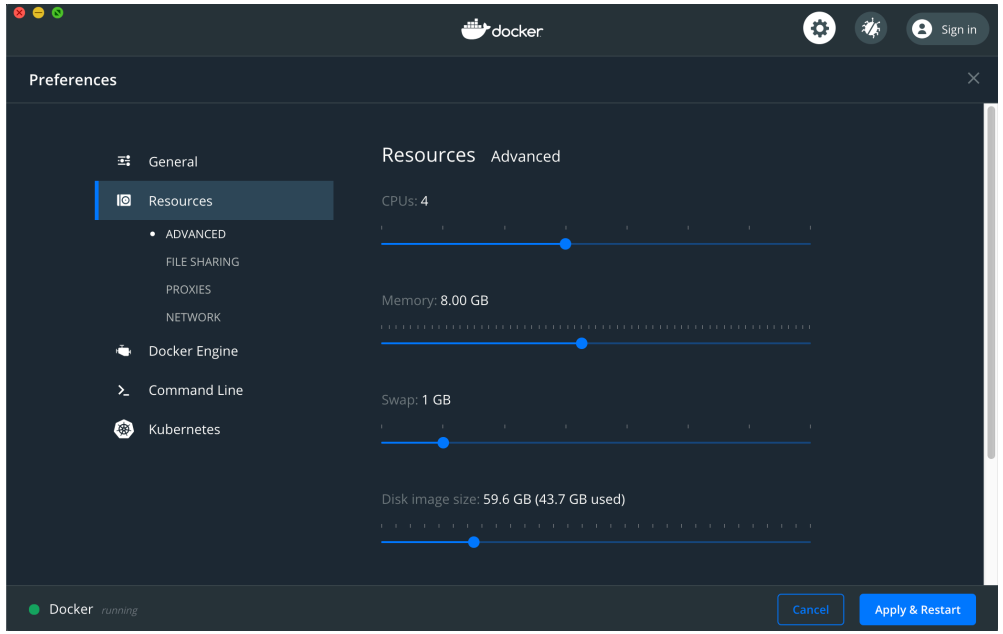
- docker-compose up –d
  - docker-compose.ymlファイルの定義に従いDockerイメージを取得・ビルドし、Dockerコンテナを作成・起動します。"--build"オプションを付与することで、イメージがすでに存在する場合も強制的にビルドします
- docker-compose down
  - docker-compose.ymlの定義に従い、関連するコンテナを削除します。"--rmi all"オプションを付与することで、関連するイメージも削除することができます。
- docker-compose stop
  - コンテナを停止します。
- docker-compose start
  - コンテナを起動します。
- docker compose exec サービス名（notebook | db） bash
  - コンテナ内のLinux OSに入ることができます。dss-notebnookに入る場合はnotebookを、dss-postgresに入る場合はdbをサービス名として指定します。

Docker Desktop設定方法

dockerの初期設定では、CPUやメモリの使用量が抑えられているケースがあります。また、自身のホームディレクトリ配下以外に100本ノックのファイル群を展開した場合、Dockerのファイル共有設定が必要となります。

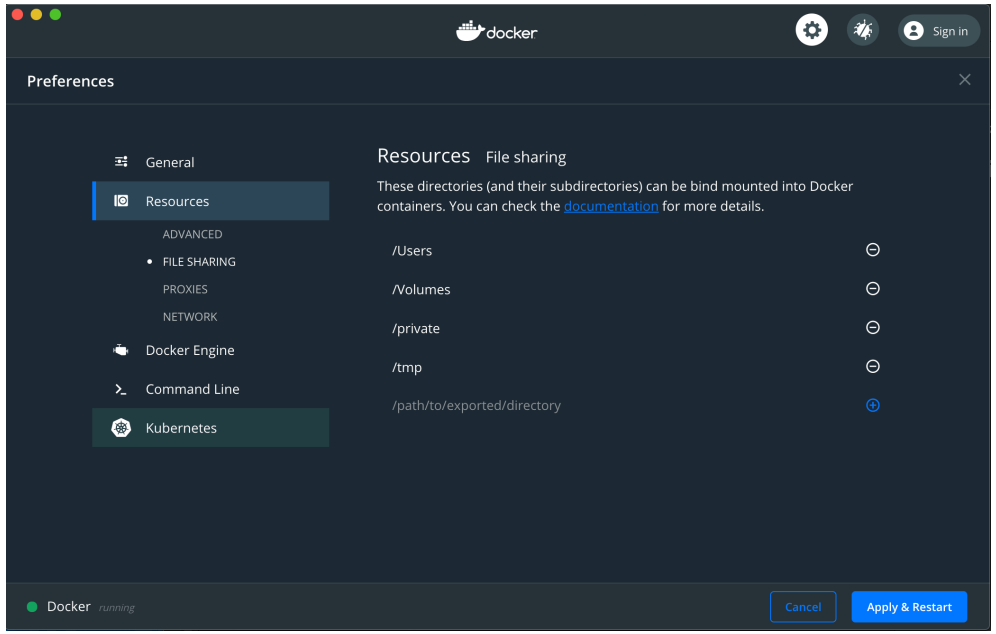
■リソース設定

Macの場合はMenuバー、Windowsの場合はタスクトレイにあるDockerアイコンをクリックし、Preference → Resource → ADVANCEDと進んで設定します。CPUとメモリそれぞれPC搭載量の半分程度を推奨します。



■共有設定

Macの場合はMenuバー、Windowsの場合はタスクトレイにあるDockerアイコンをクリックし、Preference → Resource → FILE SHARINGと進んで設定します。100本ノックファイル群を格納したディレクトリを追加します。

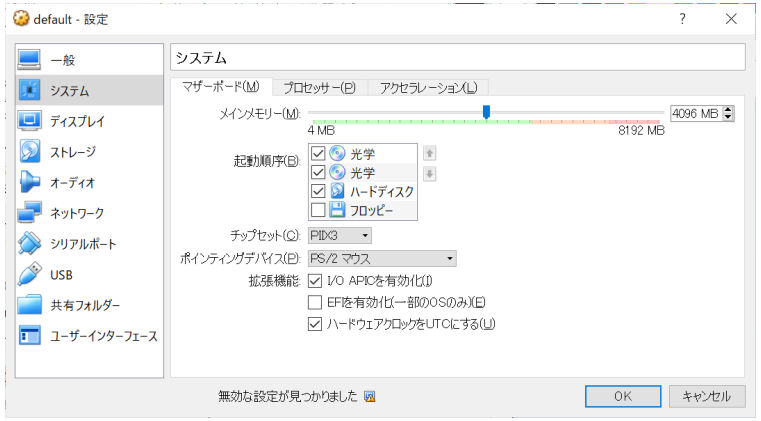


Docker Toolbox設定方法

Docker ToolboxはOracle VirtualBox上で動作します。Oracle VirtualBoxの初期設定では、CPUやメモリの使用量が抑えられています。また、自身のホームディレクトリ配下以外に100本ノックのファイル群を展開した場合、ファイル共有設定が必要となります。

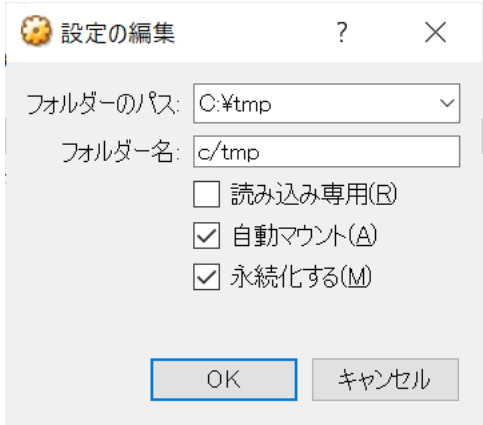
■リソース設定

WindowsのスタートメニューからOracle VirtualBoxを起動します。仮想マシンが起動中の場合は停止します。設定 → システムと進み、マザーボードタブでメモリを、プロセサータブでCPUを設定します。PC搭載量の半分程度を推奨します。設定が完了したらVirtualBoxを終了し、Docker Toolboxを起動します。



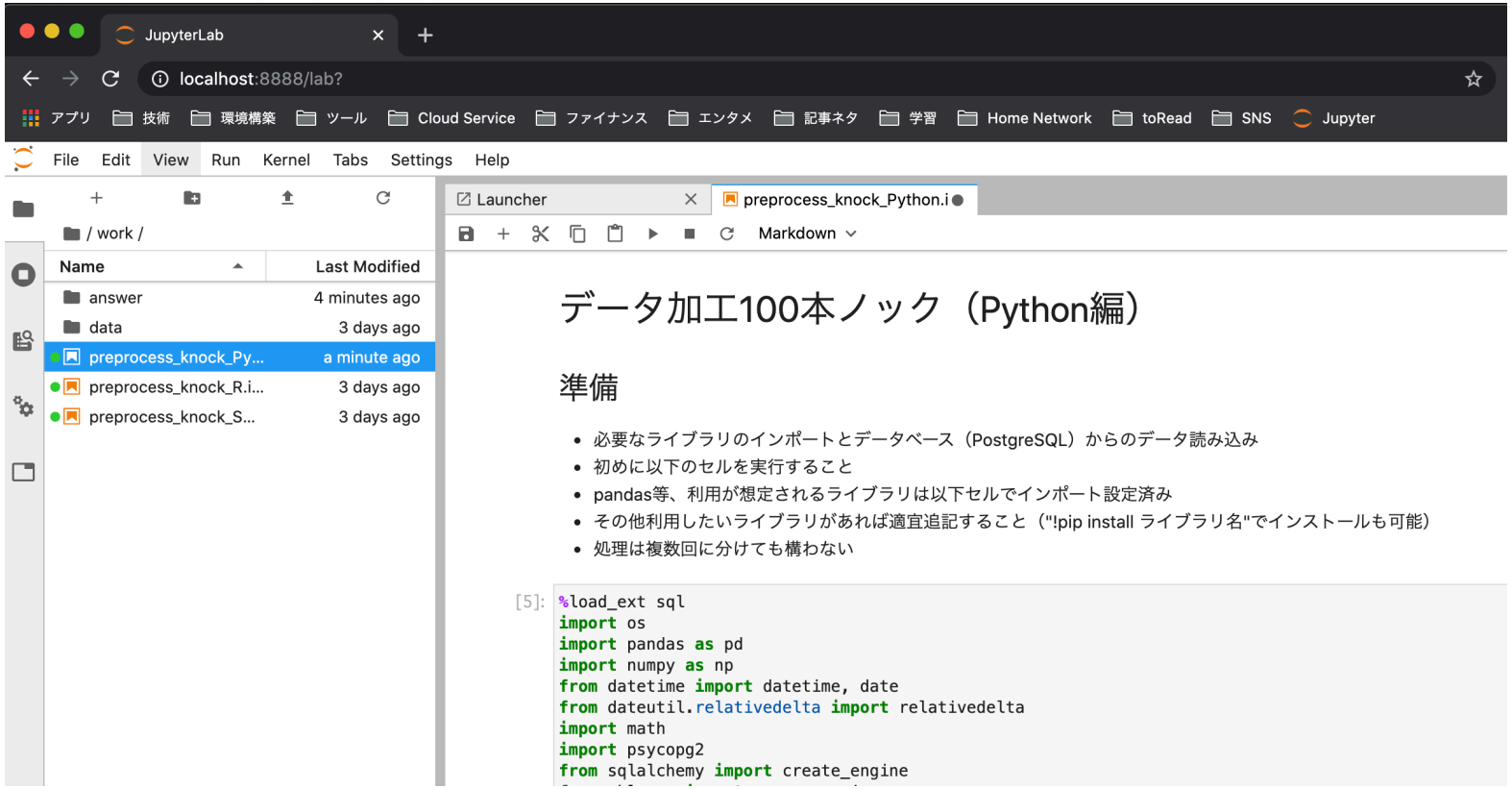
■共有設定

WindowsのスタートメニューからOracle VirtualBoxを起動します。仮想マシンが起動中の場合は停止します。設定 → 共有フォルダーと進んで設定します。100本ノックファイル群を格納したディレクトリを追加します（以下はC:¥tmp配下に格納した場合の設定）。



利用方法

Docker Desktopの場合はhttp://localhost:8888、DockerToolboxの場合はhttp://192.168.99.100:8888にブラウザからアクセスします。workフォルダ内にnotebookファイルを開いて、ノックを実践しましょう。



その他注意事項

【コンテナ作成に失敗した場合】
notebookファイルが存在しない、またはPostgreSQLのデータベースにアクセスできない、場合はコンテナの作成に失敗している可能性があります。

① 100本ノックファイル群を格納したディレクトリがマウントできていないケース
ホームディレクトリ配下以外にファイルを格納した場合、DockerやVirtual Boxのファイル共有設定が必要となります。設定を確認しましょう。

② Windows版gitが勝手に改行コードを変えてしまうケース
PostgreSQLでデータベースを初期設定するためのスクリプトなど、本環境構築のスクリプト群は文字コードUTF-8、改行コードLFで記述されていますが、Windows版gitはインストール時の設定により改行コードをCRLFに変換してしまうケースがあります。改行コードがCRLFとならないよう設定するか、GitHubからZIPファイルで取得することで対応できます。

③ コンテナのbuildでエラーとなるケース
dockerを再起動してから実行することで解消することがあります。docker-machine restart のコマンドなどでdockerを再起動してから実行してみてください。

【免責】
データサイエンス100本ノック（構造化データ加工編）の利用に関するご質問等について、個別での対応は受けかねますので予めご了承ください。
また、データサイエンス100本ノック（構造化データ加工編）の利用により生じるいかなる問題についても、当協会は一切の責任を負いかねますのであらかじめご了承ください。

参考図書

- データサイエンス100本ノック（データ加工編）を作成するにあたり、以下の書籍や各種Webサイトを参考とさせていただきました。

本橋智光[著] 株式会社ホクソエム[監修]

「前処理大全 データ分析のためのSQL/R/Python実践テクニック」 技術評論社 2018年

Alice Zheng、Amanda Casari [著] 株式会社ホクソエム[訳]

「機械学習のための特徴量エンジニアリング その原理とPythonによる実践」 技術評論社 2019年