Python and Snowflake

New Zealand Python User Group – Auckland

John Graves

2021-07-21

AGENDA

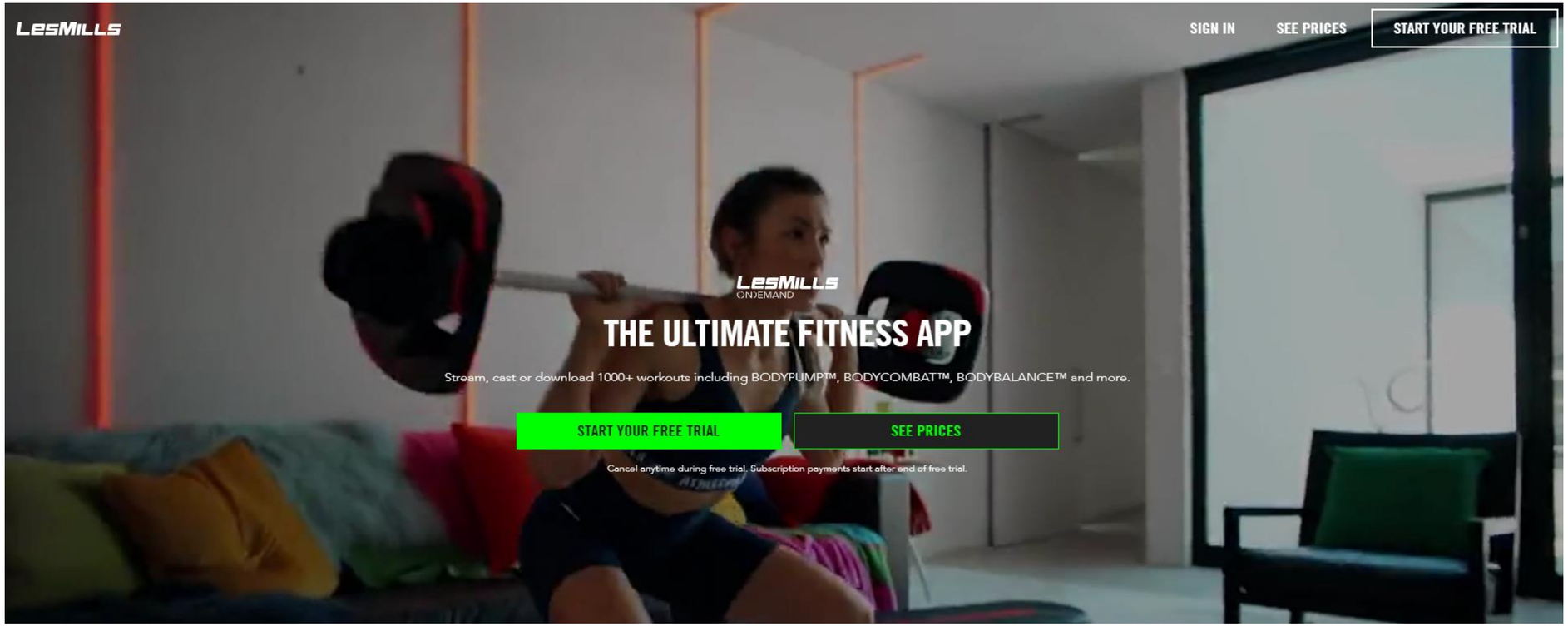
PYTHON AND SNOWFLAKE AT LES MILLS

- Introduction
- Context
- Tools
- Tips & Tricks
- Links

INTRODUCTION

JOHN GRAVES AT LES MILLS

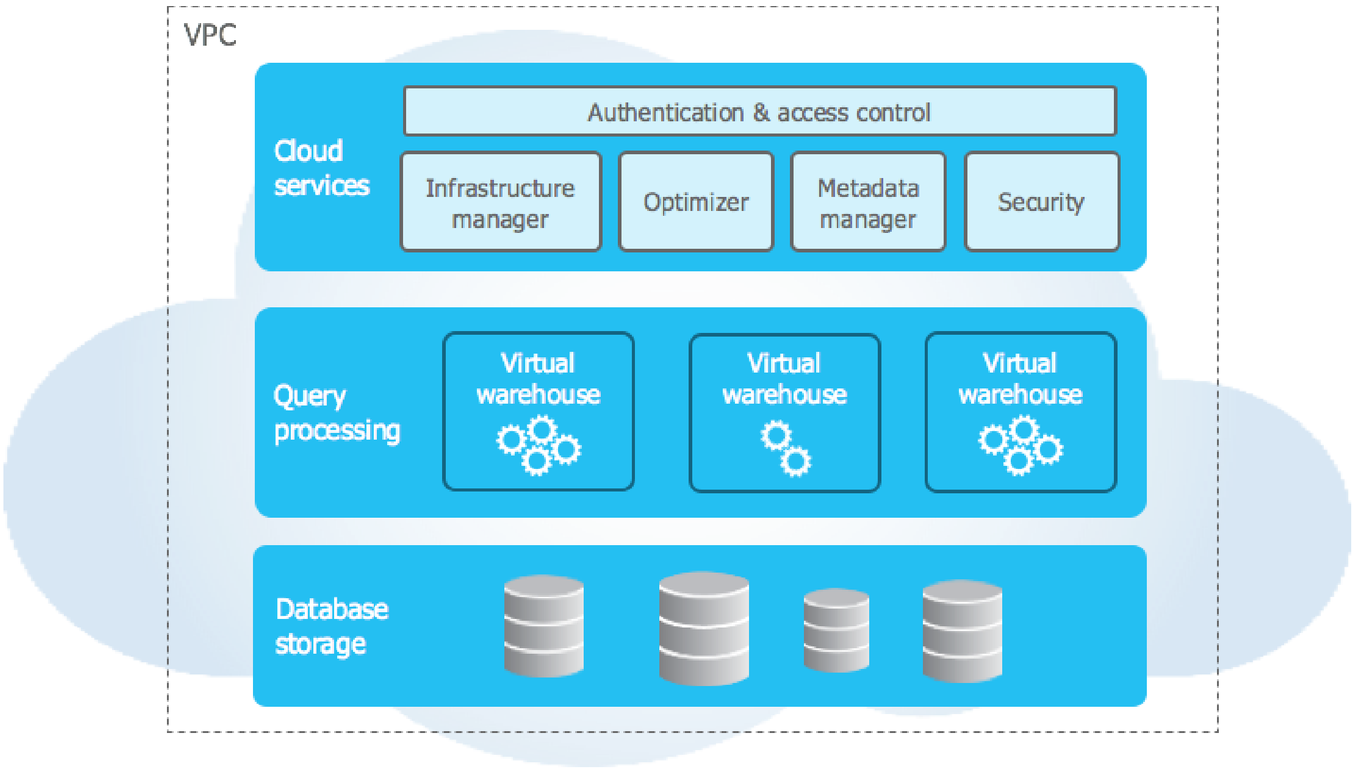
- **Who is John Graves?** Senior Data Scientist - AUT PhD
  Before Les Mills @ Datamine/Qrious/Electric Kiwi
- **What is Les Mills?**
  Prior to COVID: *International* B2B with 130,000 instructors
  Since COVID: +*On Demand* D2C



CONTEXT AND BACKGROUND

SNOWFLAKE AT LES MILLS

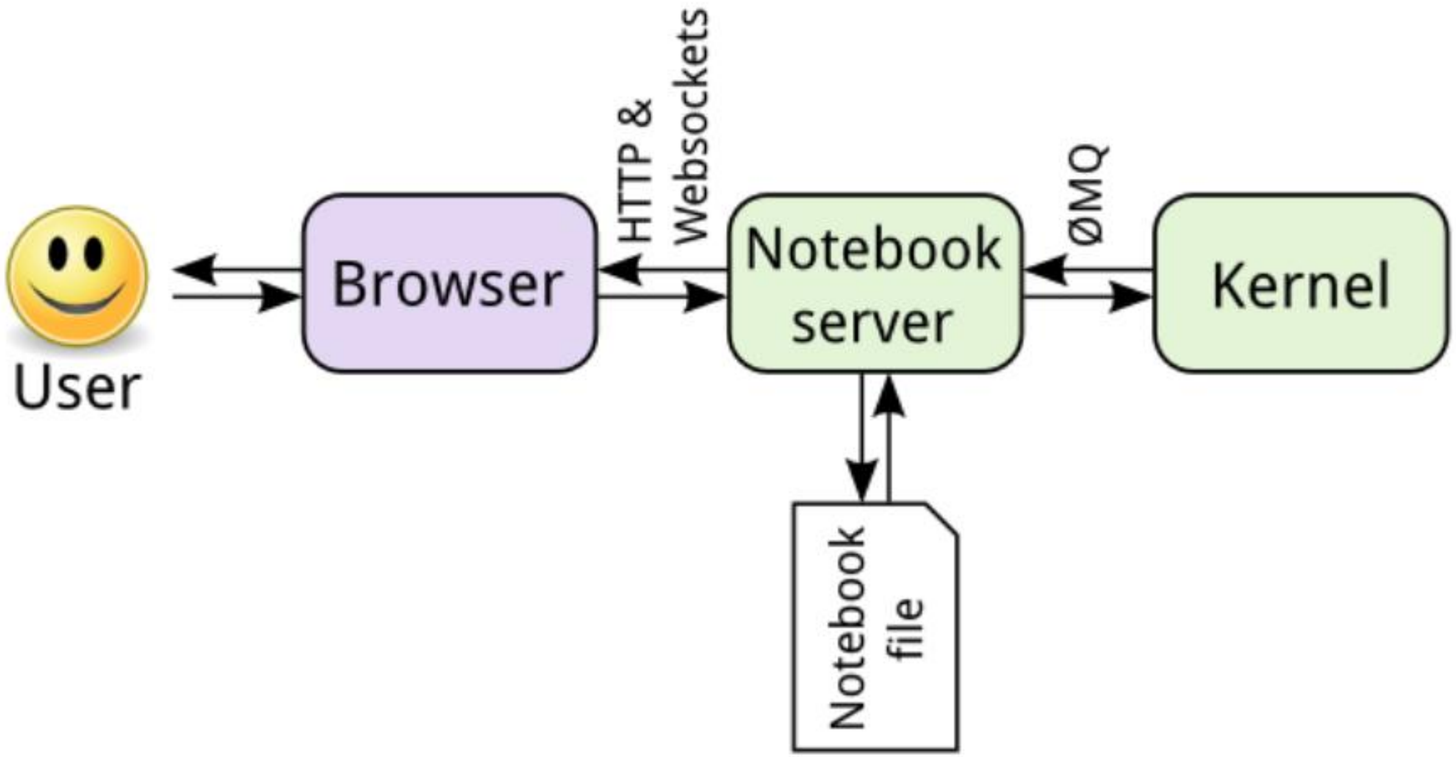
- **What is Snowflake?** Cloud-based data warehouse
- **Why Snowflake?** SQL + distributed processing (like **AWS Athena**)
- **When at Les Mills?** Since 2019 migration from on-prem Microsoft SQL Server



CONTEXT AND BACKGROUND

JUPYTER AT LES MILLS

- **What is Jupyter?** Browser-based literate programming
- **Why Jupyter?** Notebook standard for Data Science
- **When at Les Mills?**
  Since May 2021



TOOLS

WITHIN JUPYTER

- **Python** Programming language (1991)
- **Pandas** “Panel data” library (2008)
- **Plotly** Visualization library (2013)

- **sidetable** Frequency table utility (2020)

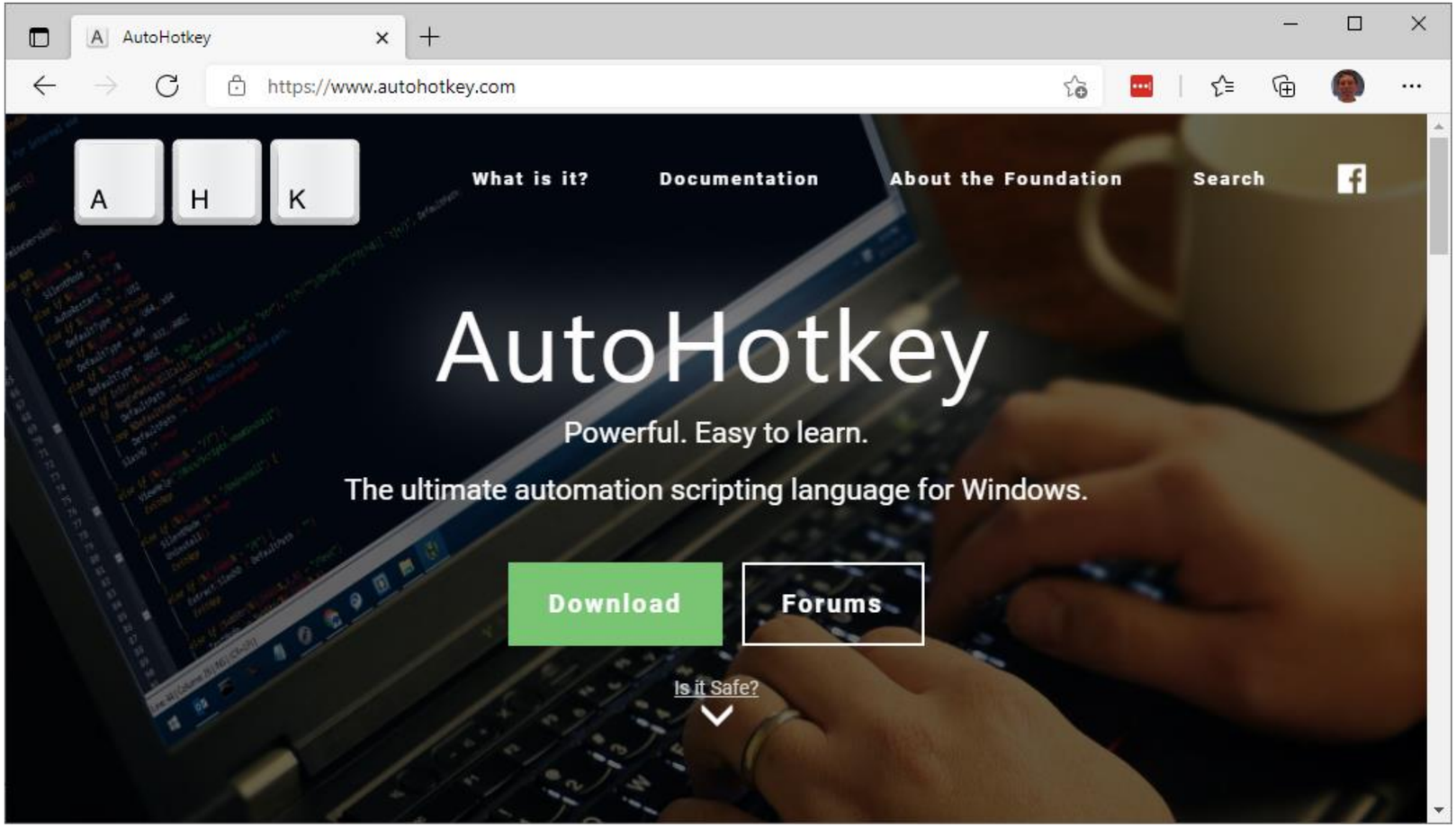
For the Titanic data: `df.stb.freq(['class'])` will build a frequency table like this:

	class	count	percent	cumulative_count	cumulative_percent
0	Third	491	55.1066	491	55.1066
1	First	216	24.2424	707	79.349
2	Second	184	20.651	891	100

TOOLS

AROUND JUPYTER

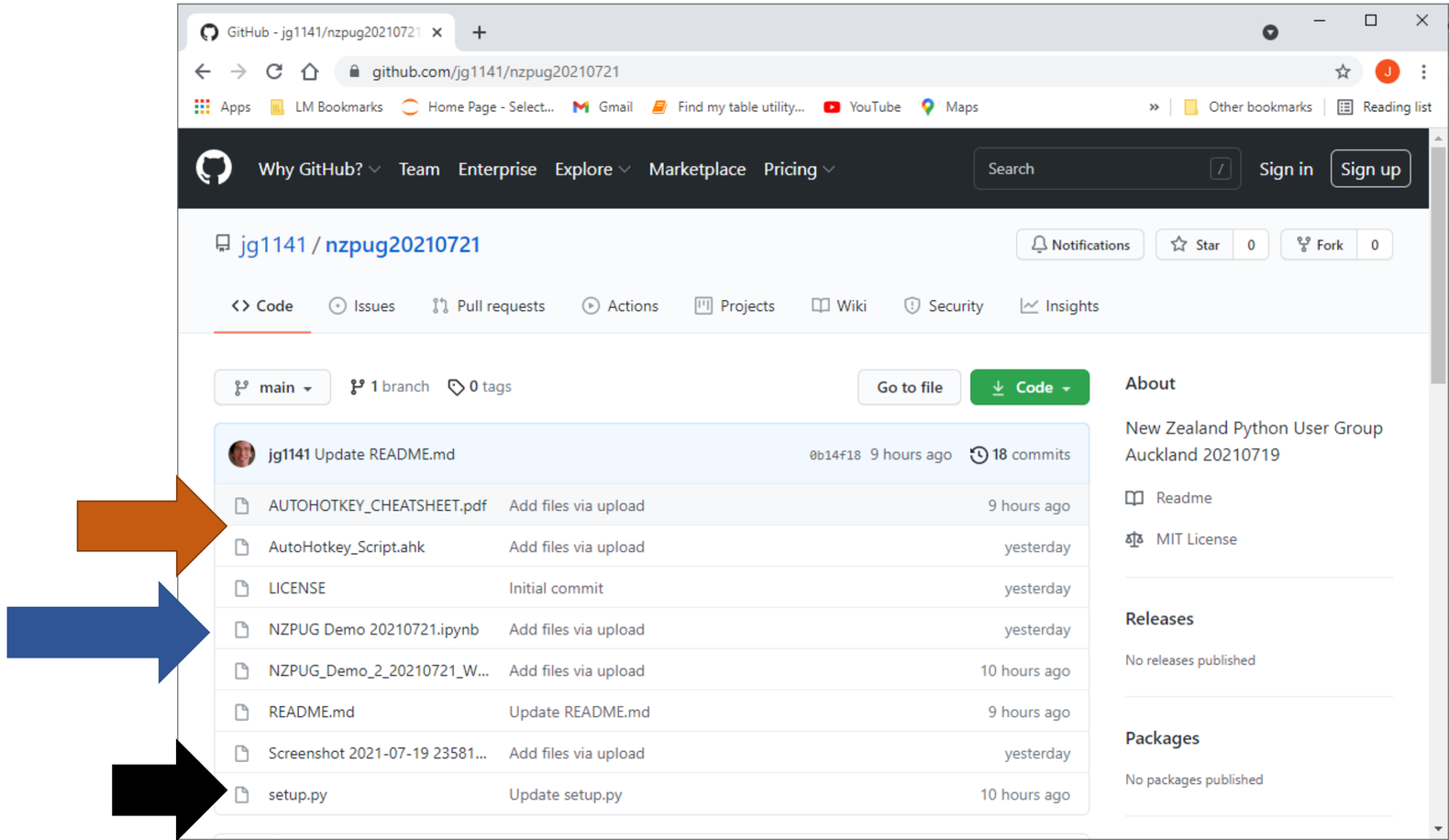
- **Snowflake Connector** Snowflake API library
- **setup.py** Notebook function initialization
- **AutoHotkey**
  Keyboard scripting

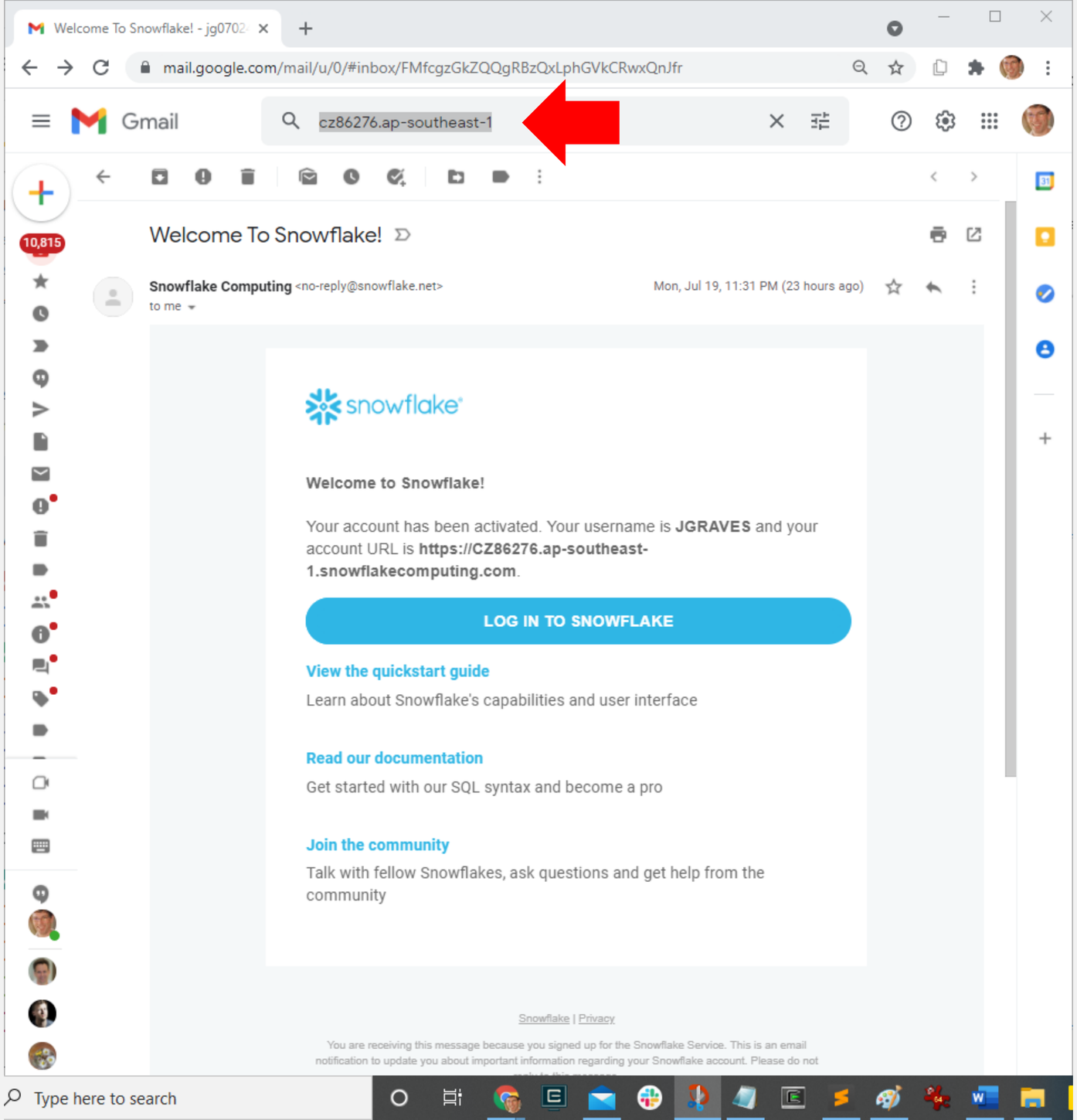


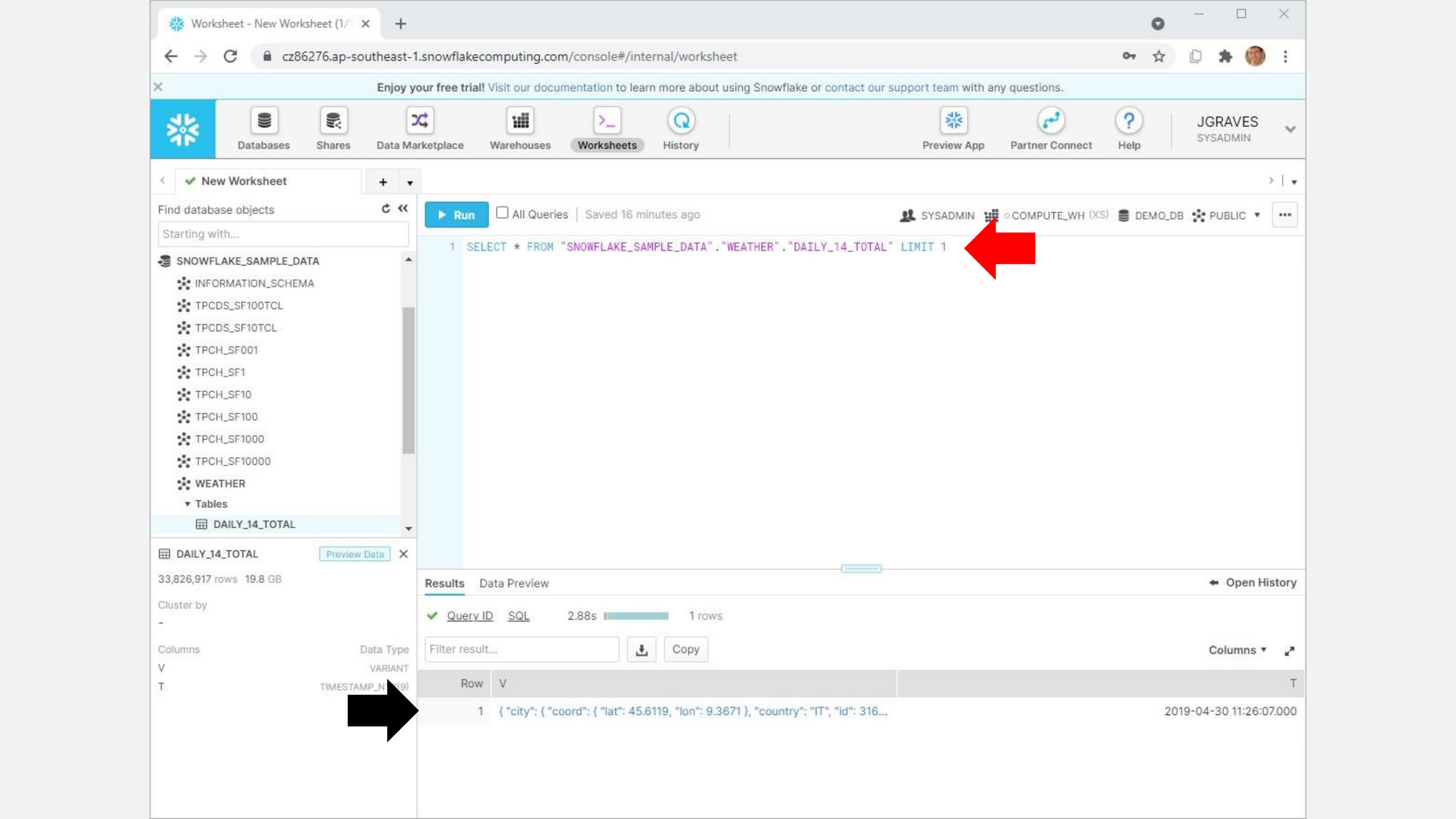
DEMO 1 –https://bit.ly/nzpug210721

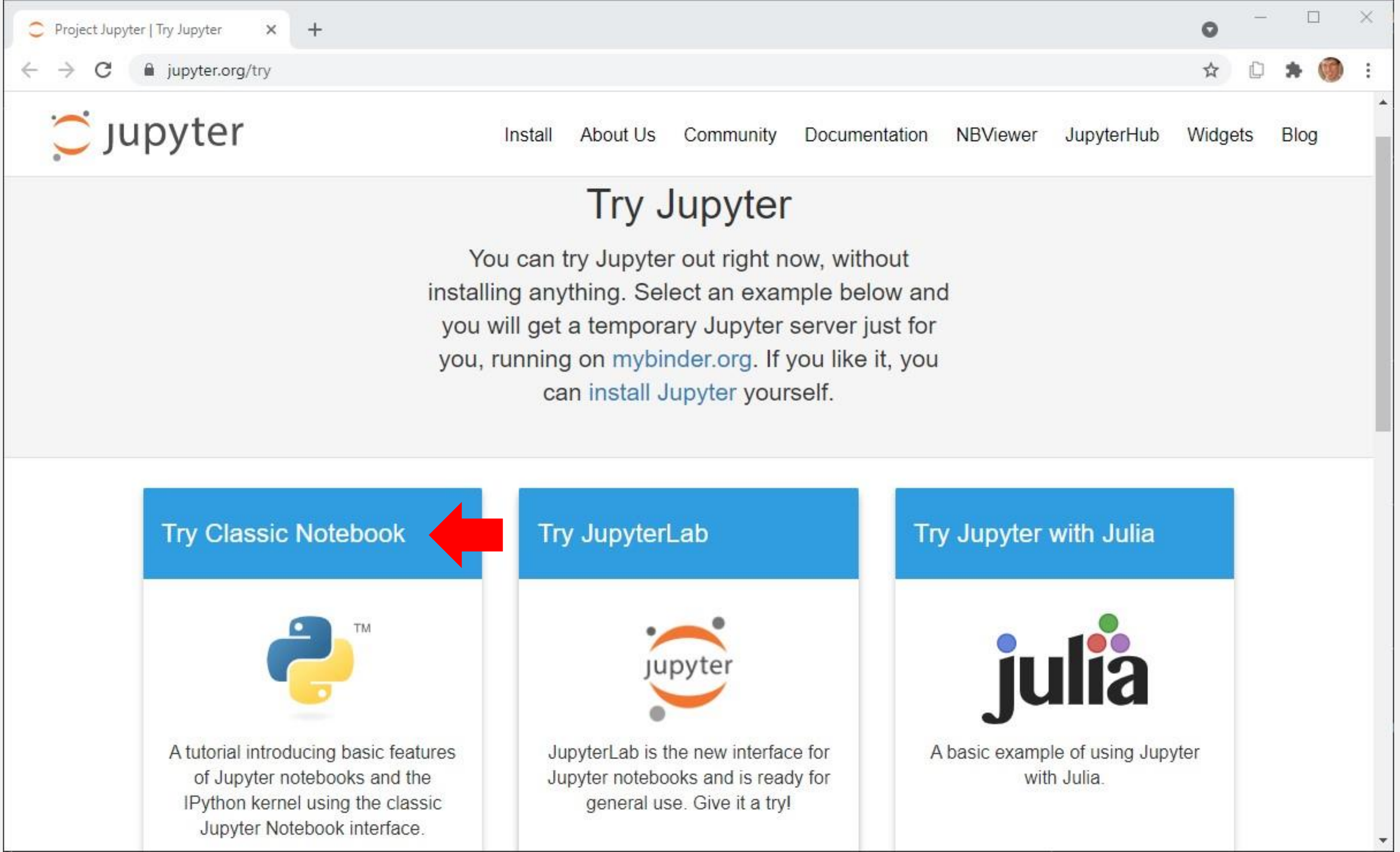
INSTALLING LIBRARIES, SETUP

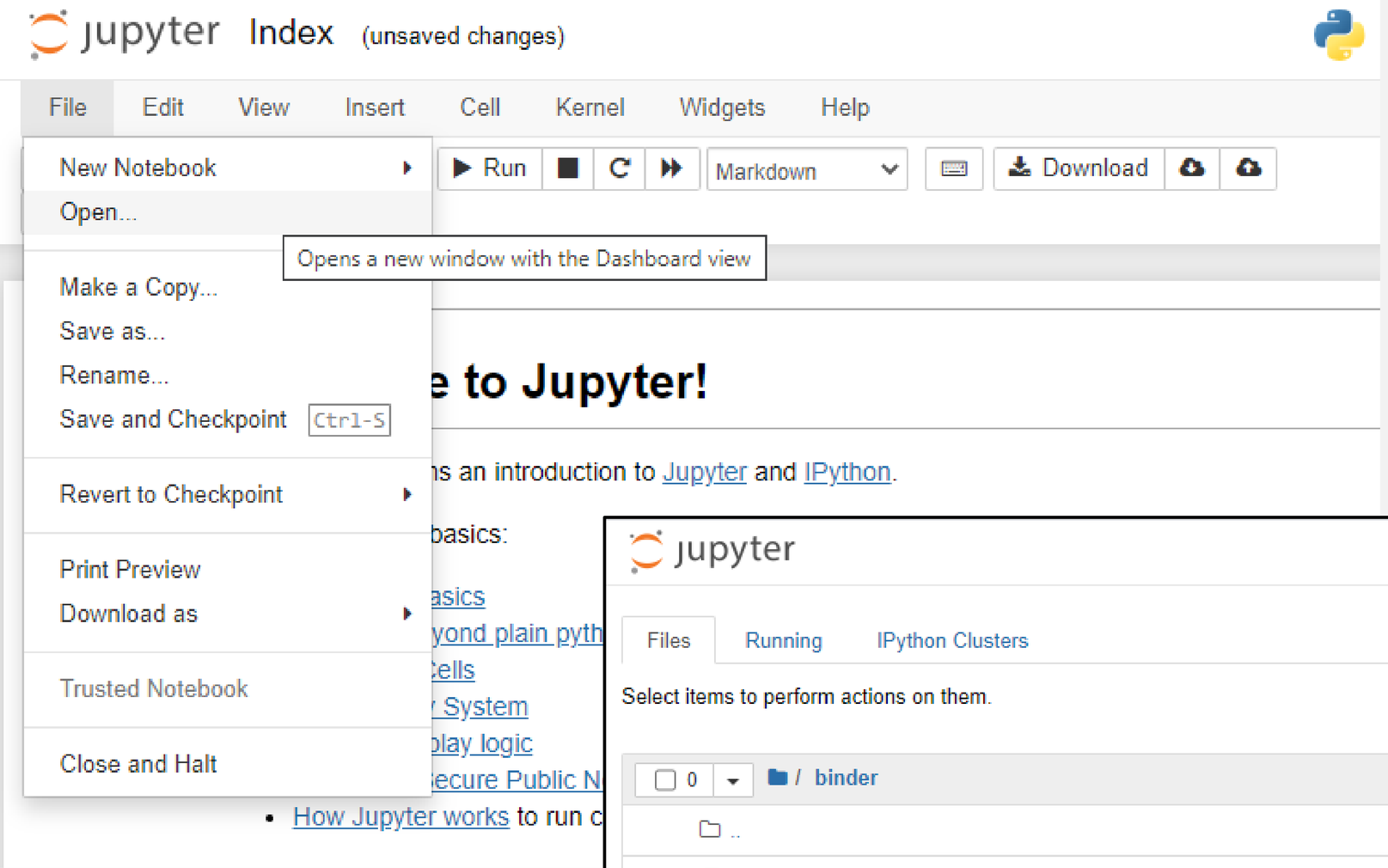
- **Snowflake Free Trial** https://signup.snowflake.com/
- **.ipynb** Demo notebook *containing* **setup.py**
- **AutoHotkey**
  Keyboard scripting

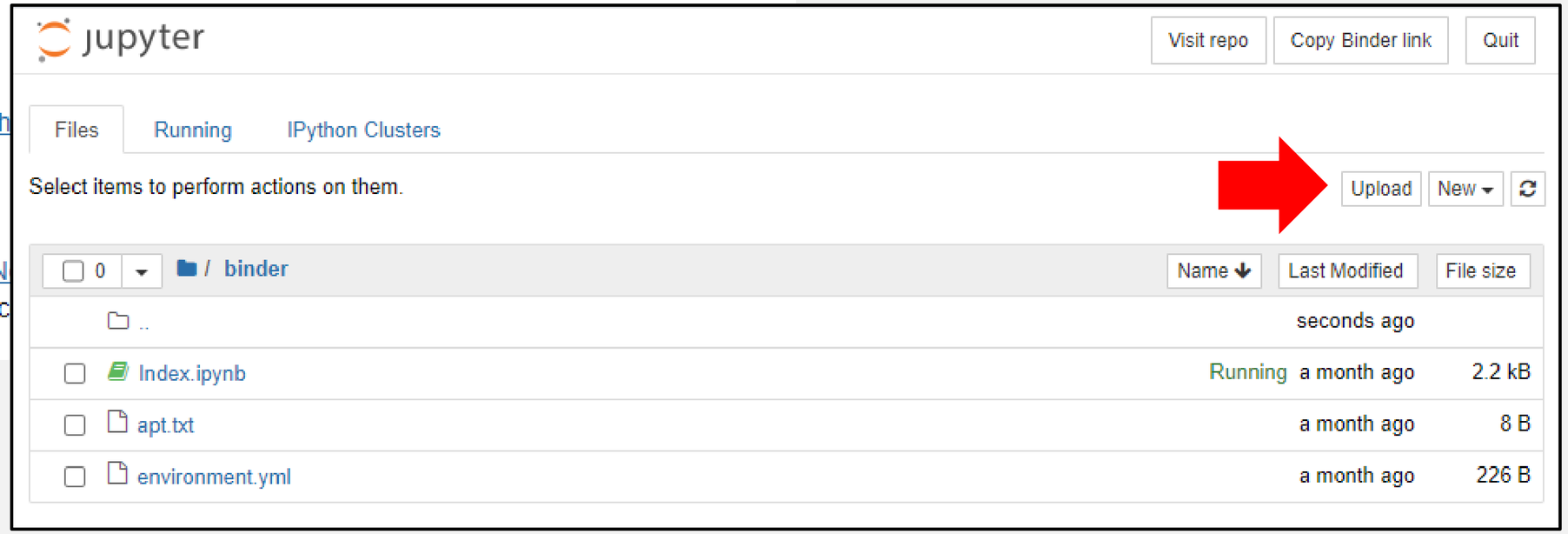



Welcome To Snowflake! - jg0702
mail.google.com/mail/u/0/#inbox/FMfcgzGkZQQgRBzQxLphGVkCRwxQnJfr
Gmail
cz86276.ap-southeast-1
10,815
Welcome To Snowflake!
Snowflake Computing <no-reply@snowflake.net>
to me
Mon, Jul 19, 11:31 PM (23 hours ago)
snowflake
Welcome to Snowflake!
Your account has been activated. Your username is JGRAVES and your account URL is https://CZ86276.ap-southeast-1.snowflakecomputing.com.
LOG IN TO SNOWFLAKE
View the quickstart guide
Learn about Snowflake's capabilities and user interface
Read our documentation
Get started with our SQL syntax and become a pro
Join the community
Talk with fellow Snowflakes, ask questions and get help from the community
Snowflake | Privacy
You are receiving this message because you signed up for the Snowflake Service. This is an email notification to update you about important information regarding your Snowflake account. Please do not
Type here to search


Worksheet - New Worksheet (1/
cz86276.ap-southeast-1.snowflakecomputing.com/console#/internal/worksheet
Enjoy your free trial! Visit our documentation to learn more about using Snowflake or contact our support team with any questions.
Databases
Shares
Data Marketplace
Warehouses
Worksheets
History
Preview App
Partner Connect
Help
JGRAVES
SYSADMIN
New Worksheet
Find database objects
Starting with...
SNOWFLAKE_SAMPLE_DATA
INFORMATION_SCHEMA
TPCDS_SF100TCL
TPCDS_SF10TCL
TPCH_SF001
TPCH_SF1
TPCH_SF10
TPCH_SF100
TPCH_SF1000
TPCH_SF10000
WEATHER
Tables
DAILY_14_TOTAL
Run
All Queries
Saved 16 minutes ago
SYSADMIN
COMPUTE_WH (XS)
DEMO_DB
PUBLIC
1 SELECT * FROM "SNOWFLAKE_SAMPLE_DATA"."WEATHER"."DAILY_14_TOTAL" LIMIT 1
DAILY_14_TOTAL
Preview Data
33,826,917 rows 19.8 GB
Cluster by
-
Columns
Data Type
V
VARIANT
T
Results
Data Preview
Open History
Query ID
SQL
2.88s
1 rows
Filter result...
Copy
Columns
Row
V
T
1
{ "city": { "coord": { "lat": 45.6119, "lon": 9.3671 }, "country": "IT", "id": 316...
2019-04-30 11:26:07.000


Project Jupyter | Try Jupyter
jupyter.org/try
jupyter
Install
About Us
Community
Documentation
NBViewer
JupyterHub
Widgets
Blog
Try Jupyter
You can try Jupyter out right now, without installing anything. Select an example below and you will get a temporary Jupyter server just for you, running on mybinder.org. If you like it, you can install Jupyter yourself.
Try Classic Notebook
TM
A tutorial introducing basic features of Jupyter notebooks and the IPython kernel using the classic Jupyter Notebook interface.
Try JupyterLab
jupyter
JupyterLab is the new interface for Jupyter notebooks and is ready for general use. Give it a try!
Try Jupyter with Julia
julia
A basic example of using Jupyter with Julia.


jupyter Index (unsaved changes)
File Edit View Insert Cell Kernel Widgets Help
New Notebook
Open...
Make a Copy...
Save as...
Rename...
Save and Checkpoint Ctrl-S
Revert to Checkpoint
Print Preview
Download as
Trusted Notebook
Close and Halt
Run
Markdown
Download
Opens a new window with the Dashboard view
e to Jupyter!
ns an introduction to Jupyter and IPython.
basics:
asics
yond plain pyth
Cells
System
play logic
ecure Public N
How Jupyter works to run c


jupyter
Visit repo
Copy Binder link
Quit
Files
Running
IPython Clusters
Select items to perform actions on them.
Upload
New
0
/ binder
Name
Last Modified
File size
..
seconds ago
Index.ipynb
Running
a month ago
2.2 kB
apt.txt
a month ago
8 B
environment.yml
a month ago
226 B

NZPUG Demo 20210719 - Jupyte

hub.gke2.mybinder.org/user/ipython-ipython-in-depth-7b798mlm/notebooks/binder/NZPUG%20Demo%2020210719.ipynb#

jupyter NZPUG Demo 20210721

Visit repo | Copy Binder link

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3

Run | Code | Download | GitHub | Binder

Memory: 259.3 MB / 2 GB

In [1]:
```
!python --version
```

```
Python 3.6.13
```

In [2]:
```
!pip install humanize numpy pandas plotly pyperclip sidetable
```

```
Requirement already satisfied: humanize in /srv/conda/envs/notebook/lib/python3.6/site-packages (3.10.0)
Requirement already satisfied: numpy in /srv/conda/envs/notebook/lib/python3.6/site-packages (1.19.5)
Requirement already satisfied: pandas in /srv/conda/envs/notebook/lib/python3.6/site-packages (1.1.5)
Requirement already satisfied: plotly in /srv/conda/envs/notebook/lib/python3.6/site-packages (5.1.0)
Requirement already satisfied: pyperclip in /srv/conda/envs/notebook/lib/python3.6/site-packages (1.8.2)
Requirement already satisfied: sidetable in /srv/conda/envs/notebook/lib/python3.6/site-packages (0.8.0)
Requirement already satisfied: setuptools in /srv/conda/envs/notebook/lib/python3.6/site-packages (from humanize) (49.6.0.post2
0210108)
Requirement already satisfied: python-dateutil>=2.7.3 in /srv/conda/envs/notebook/lib/python3.6/site-packages (from pandas) (2.
8.1)
Requirement already satisfied: pytz>=2017.2 in /srv/conda/envs/notebook/lib/python3.6/site-packages (from pandas) (2020.1)
Requirement already satisfied: six>=1.5 in /srv/conda/envs/notebook/lib/python3.6/site-packages (from python-dateutil>=2.7.3->p
andas) (1.15.0)
Requirement already satisfied: tenacity>=6.2.0 in /srv/conda/envs/notebook/lib/python3.6/site-packages (from plotly) (8.0.1)
```

In [3]:
```
!pip install python-dotenv
```

```
Requirement already satisfied: python-dotenv in /srv/conda/envs/notebook/lib/python3.6/site-packages (0.18.0)
```

In [4]:
```
!pip install -r https://raw.githubusercontent.com/snowflakedb/snowflake-connector-python/v2.4.6/tested_requirements/requirements_
```

```
Requirement already satisfied: asn1crypto==1.4.0 in /srv/conda/envs/notebook/lib/python3.6/site-packages (from -r https://raw.g
```

NZPUG Demo 20210719 - Jupyte

hub.gke2.mybinder.org/user/ipython-ipython-in-depth-7b798mlm/notebooks/binder/NZPUG%20Demo%2020210719.ipynb#

jupyter NZPUG Demo 20210721

Visit repo | Copy Binder link

File Edit View Insert Cell Kernel Widgets Help

No kernel | Trusted | Python 3

Run | Code | Download | GitHub | Binder

Memory: 79.2 MB / 2 GB

In [5]:
```
!pip install pyarrow
```

```
Requirement already satisfied: pyarrow in /srv/conda/envs/notebook/lib/python3.6/site-packages (3.0.0)
Requirement already satisfied: numpy>=1.16.6 in /srv/conda/envs/notebook/lib/python3.6/site-packages (from pyarrow) (1.19.5)
```

In [6]:
```
!pip install snowflake-connector-python==2.4.6
```

```
Requirement already satisfied: snowflake-connector-python==2.4.6 in /srv/conda/envs/notebook/lib/python3.6/site-packages (2.4.
6)
Requirement already satisfied: setuptools>34.0.0 in /srv/conda/envs/notebook/lib/python3.6/site-packages (from snowflake-connec
tor-python==2.4.6) (49.6.0.post20210108)
Requirement already satisfied: cryptography<4.0.0,>=2.5.0 in /srv/conda/envs/notebook/lib/python3.6/site-packages (from snowfla
ke-connector-python==2.4.6) (2.9.2)
Requirement already satisfied: oscrypto<2.0.0 in /srv/conda/envs/notebook/lib/python3.6/site-packages (from snowflake-connector
-python==2.4.6) (1.2.1)
Requirement already satisfied: idna<3,>=2.5 in /srv/conda/envs/notebook/lib/python3.6/site-packages (from snowflake-connector-p
ython==2.4.6) (2.10)
Requirement already satisfied: pyOpenSSL<21.0.0,>=16.2.0 in /srv/conda/envs/notebook/lib/python3.6/site-packages (from snowflak
e-connector-python==2.4.6) (19.1.0)
Requirement already satisfied: pycryptodomex!=3.5.0,<4.0.0,>=3.2 in /srv/conda/envs/notebook/lib/python3.6/site-packages (from
snowflake-connector-python==2.4.6) (3.9.8)
Requirement already satisfied: cffi<2.0.0,>=1.9 in /srv/conda/envs/notebook/lib/python3.6/site-packages (from snowflake-connect
or-python==2.4.6) (1.14.3)
Requirement already satisfied: chardet<5,>=3.0.2 in /srv/conda/envs/notebook/lib/python3.6/site-packages (from snowflake-connec
tor-python==2.4.6) (3.0.4)
Requirement already satisfied: azure-storage-blob<13.0.0,>=12.0.0 in /srv/conda/envs/notebook/lib/python3.6/site-packages (from
snowflake-connector-python==2.4.6) (12.5.0)
Requirement already satisfied: pytz in /srv/conda/envs/notebook/lib/python3.6/site-packages (from snowflake-connector-python==
2.4.6) (2020.1)
Requirement already satisfied: asn1crypto<2.0.0,>0.24.0 in /srv/conda/envs/notebook/lib/python3.6/site-packages (from snowflake
-connector-python==2.4.6) (1.4.0)
```

NZPUG Demo 20210719 - Jupyte

hub.gke2.mybinder.org/user/ipython-ipython-in-depth-7b798mlm/notebooks/binder/NZPUG%20Demo%2020210719.ipynb#

jupyter NZPUG Demo 20210721

Visit repo | Copy Binder link

File Edit View Insert Cell Kernel Widgets Help

No kernel | Trusted | Python 3

Run | Code | Download | GitHub | Binder

Memory: 79.2 MB / 2 GB

In [7]:
```
# setup.py
# Created 20210524 1537
# Invoke with %run "C:\\setup.py"
# Modified (see version)

VERSION = "20210708 0933 "

import datetime
import humanize
import numpy as np
import os
import pandas as pd
import plotly.express as px
import pyperclip
import re
import sidetable
import snowflake.connector
import time
from snowflake.connector.pandas_tools import write_pandas
from dotenv import load_dotenv

# _ = load_dotenv()
```

In [8]:
```
# Get non-null counts
pd.options.display.max_info_rows = 16907850

# Connection string
conn = snowflake.connector.connect(
#                 user=os.getenv('user'),
#                 password=os.getenv('password'),
#                 account=os.getenv('account'),
```

NZPUG Demo 20210719 - Jupyte

hub.gke2.mybinder.org/user/ipython-ipython-in-depth-7b798mlm/notebooks/binder/NZPUG%20Demo%2020210719.ipynb#

jupyter NZPUG Demo 20210721

Visit repo | Copy Binder link

File Edit View Insert Cell Kernel Widgets Help

No kernel | Trusted | Python 3

Run | Code | Download | GitHub | Binder

Memory: 79.2 MB / 2 GB

In [8]:
```
# Get non-null counts
pd.options.display.max_info_rows = 16907850

# Connection string
conn = snowflake.connector.connect(
#                 user=os.getenv('user'),
#                 password=os.getenv('password'),
#                 account=os.getenv('account'),
#                 warehouse=os.getenv('warehouse'),
#                 database=os.getenv('database'),
#                 schema=os.getenv('schema')
                user='jgraves',
                password='',
                account='CZ86276.ap-southeast-1',
                warehouse='COMPUTE_WH',
                database='DEMO_DB',
                schema='PUBLIC'
                )
```

In [9]:
```
# Execute a statement that will generate a result set.
cur = conn.cursor()

def compare_sets(list1, list2):
    """Make a count of the intersections of two sets, A and B"""
    set1 = set(list1)
    set2 = set(list2)
    set2_intersection_set1 = set2.intersection(set1)
    result = {'IN A':[len(set1), len(set2_intersection_set1), round(len(set1)/len(set1)*100,1), round(len(set2_intersection_set1)
    result['IN B'] = [len(set2_intersection_set1), len(set2), round(len(set2_intersection_set1)/len(set1)*100,1), round(len(set2)
    result['NOT IN A'] = [0, len(set2 - set1), 0, round(len(set2 - set1)/len(set2)*100,1)]
    result['NOT IN B'] = [len(set1 - set2), 0, round(len(set1 - set2)/len(set1)*100,1), 0]
```

NZPUG Demo 20210719 - Jupyte

hub.gke2.mybinder.org/user/ipython-ipython-in-depth-7b798mlm/notebooks/binder/NZPUG%20Demo%2020210719.ipynb#

jupyter NZPUG Demo 20210721

Visit repo | Copy Binder link

File Edit View Insert Cell Kernel Widgets Help

No kernel | Trusted | Python 3

Run | Code | Download | GitHub | Binder

Memory: 79.2 MB / 2 GB

In [9]:

```
# Execute a statement that will generate a result set.
cur = conn.cursor()

def compare_sets(list1, list2):
    """Make a count of the intersections of two sets, A and B"""
    set1 = set(list1)
    set2 = set(list2)
    set2_intersection_set1 = set2.intersection(set1)
    result = {'IN A':[len(set1), len(set2_intersection_set1), round(len(set1)/len(set1)*100,1), round(len(set2_intersection_set1)
    result['IN B'] = [len(set2_intersection_set1), len(set2), round(len(set2_intersection_set1)/len(set1)*100,1), round(len(set2)
    result['NOT IN A'] = [0, len(set2 - set1), 0, round(len(set2 - set1)/len(set2)*100,1)]
    result['NOT IN B'] = [len(set1 - set2), 0, round(len(set1 - set2)/len(set1)*100,1), 0]
    df = pd.DataFrame.from_dict(result, orient='index', columns=['A', 'B', '% of A', '% of B'])
    return df


def d(vars):
    """List of variables starting with string "df" in reverse order. Usage: d(dir())

    @vars list of variables output by dir() command
    """
    list_of_dfs = [item for item in vars if (item.find('df') == 0 and item.find('_') == -1 and item != 'dfs')]
    list_of_dfs.sort(key=lambda x:int(re.sub("[^0-9]", "", x.replace('df',''))) if len(x) > 2 else 0, reverse=True)
    return list_of_dfs


def e(start_time):
    """Return human readable time delta

    @start_time time to compare to current time
    """
    print(f'Time now: {datetime.datetime.now().strftime("%Y-%m-%d %H:%M")}')
```

```
def execute(sql):
    """Execute a SQL command"""
    start_time = time.monotonic()
    _ = cur.execute(sql)
    end_time = time.monotonic()
    elapsed = end_time - start_time
    print(f"Elapsed time {elapsed:.2f}")
    return


def find_col_with(df, char_to_find):
    """Return column index of first column containing char_to_find

    @char_to_find character to search for in column name
    """
    first_column_with_char_to_find = [col for col in df.columns if col.find(char_to_find) > -1][0]
    return list(df.columns).index(first_column_with_char_to_find)


def find_max_order(df, start_col=1):
    """Find the max value in each column and use it to put columns in rank order

    @start_col Index of starting column (typically 1 as first column -- column 0 -- is a date or label)
    """
    return list(df[df.columns[start_col:]].max().sort_values(ascending=False).keys())


def find_percentage_total(df, start_col=1):
    """Find total and percent of total for columns of Pandas dataframe

    @start_col Index of starting column (typically 1 as first column -- column 0 -- is a date or label)
    """
    # Get values for col1,col2 and col3
    total = pd.Series(data=np.zeros(len(df)))
    col_count = len(df.columns)
    for i in range(start_col, col_count):
```

```
        pct_of_total = round((df.iloc[:,i]/total)*100, 2)

        # Create Pandas DF with new column of pct_of_total
        df.insert(len(df.columns),f"{df.columns[i]} %", pct_of_total)

    # Pull original dataframe to show total and %
    return df


def query(sql):
    """Run a SQL query and fetch result into Pandas DataFrame"""
    start_time = time.monotonic()
    _ = cur.execute(sql)
    df = cur.fetch_pandas_all()
    end_time = time.monotonic()
    elapsed = end_time - start_time
    print(f"Elapsed time {elapsed:.2f}")
    return df


def t(title_string):
    """Add "as at {today}" to title. Usage: t(title_sting)

    @title_string text to preceed the "as at" part
    """
    today = datetime.datetime.today().strftime('%d %b %Y')
    title = f"{title_string} as at {today}"
    print(title)
    pyperclip.copy(title)
    print("(now on clipboard)")
    return title


start_time = time.monotonic()
print(f"Setup Complete v {VERSION}")
```

```
Setup Complete v 20210708 0933
```

NZPUG Demo 20210721 - Jupyte

hub.gke2.mybinder.org/user/ipython-ipython-in-depth-7b798mlm/notebooks/binder/NZPUG%20Demo%2020210721.ipynb

jupyter NZPUG Demo 20210721

Visit repo | Copy Binder link

File Edit View Insert Cell Kernel Widgets Help

No kernel | Trusted | Python 3

Run | Code | Download | GitHub | Binder

Memory: 79.2 MB / 2 GB

```
    today = datetime.datetime.today().strftime("%d %b %Y")
    title = f"{title_string} as at {today}"
    print(title)
    pyperclip.copy(title)
    print("(now on clipboard)")
    return title


start_time = time.monotonic()
print(f"Setup Complete v {VERSION}")
```

```
Setup Complete v 20210708 0933
```

In [10]:

```
df = query("""SELECT * FROM "SNOWFLAKE_SAMPLE_DATA"."WEATHER"."DAILY_14_TOTAL" LIMIT 1
""")
df
```

```
Elapsed time 0.29
```

Out[10]:

	V	T
0	{\n "city": {\n "coord": {\n "lat": 4...	2019-05-06 23:19:29

In [11]:

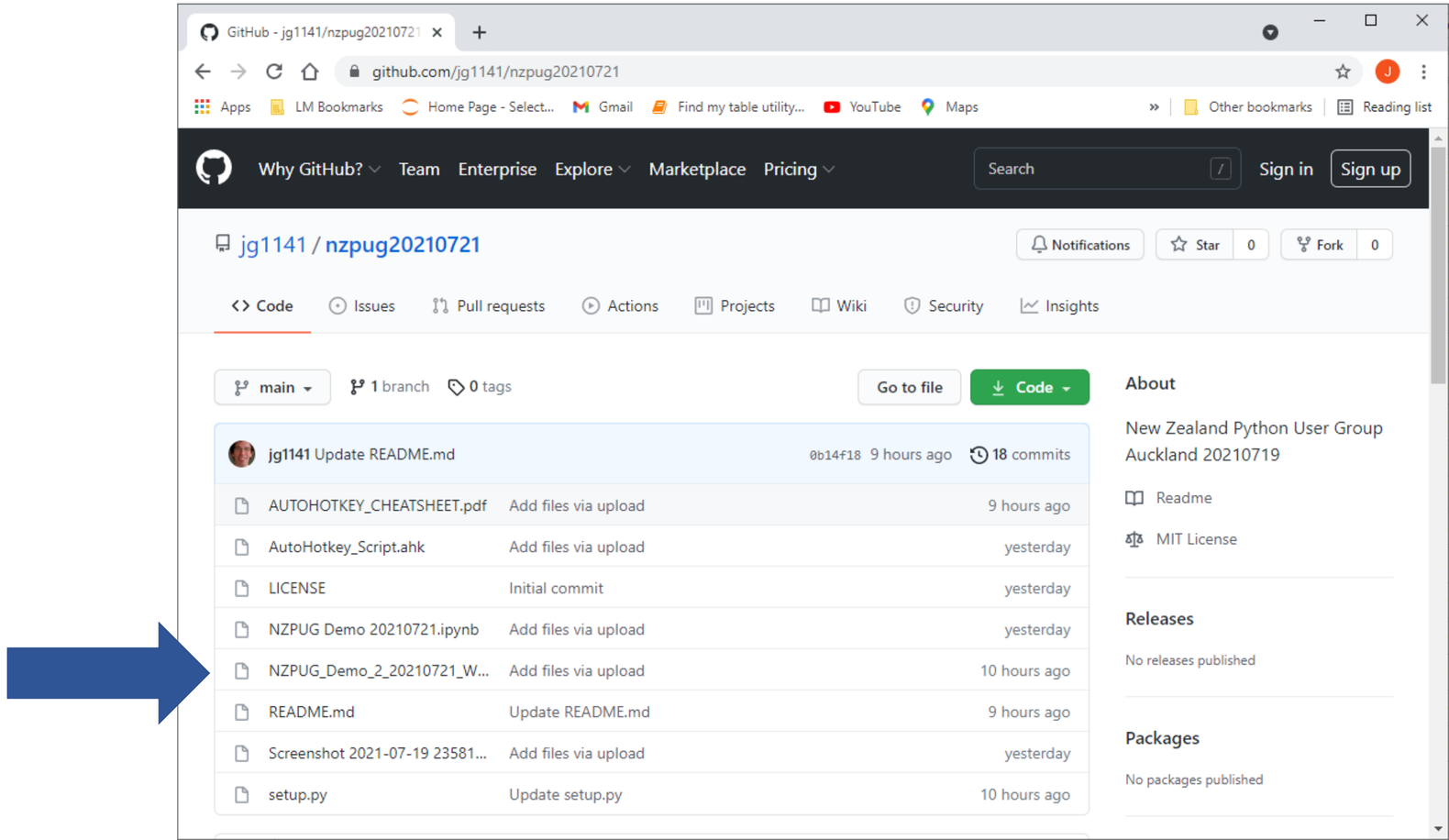
```
e(start_time)
```

```
Time now: 2021-07-19 11:53
Time since start: 0 seconds
```

DEMO 2 –https://bit.ly/nzpug210721

WRITING PANDAS DATAFRAME TO SNOWFLAKE

- **.ipynb** Demo 2 notebook *containing* **setup.py**



```
[6] df = pd.DataFrame.from_dict({"a":[1,2,3], "b":[4,5,6]})
    df
```

	a	b
0	1	4
1	2	5
2	3	6

```
[7] sql = "SELECT "
    column_list = []
    for col in df.columns:
      column_list.append(f'''{df[col].max()} as "{col}"''')
    sql += ", ".join(column_list)
    sql
```

```
'SELECT 3 as "a", 6 as "b"'
```

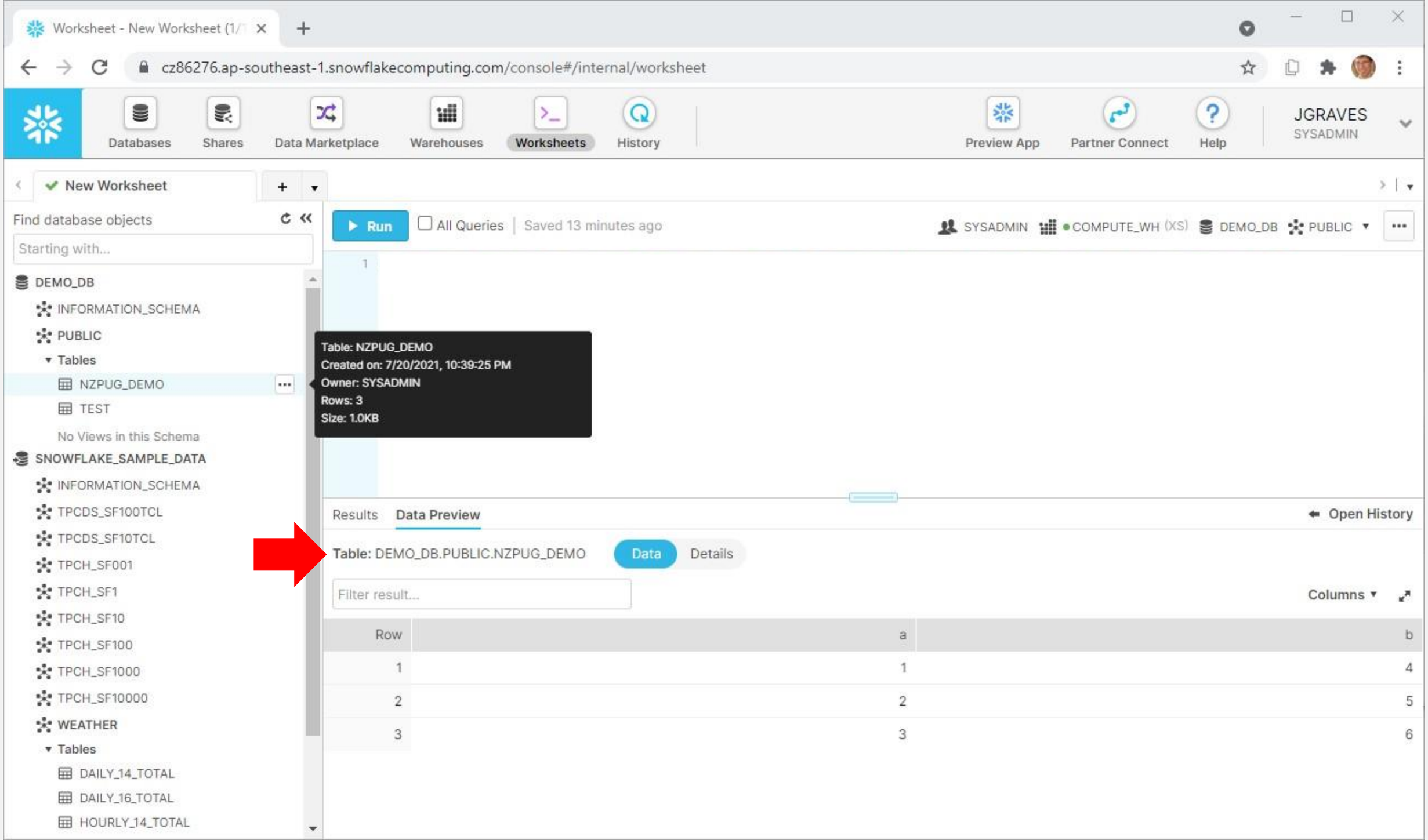
```
TABLE = "NZPUG_DEMO"
execute(f"""DROP TABLE IF EXISTS {TABLE}""")
execute(f"""CREATE TABLE {TABLE} AS ({sql})""")
execute(f"""TRUNCATE TABLE IF EXISTS {TABLE}""")
write_pandas(conn, df, TABLE)
```

```
Elapsed time 0.28
Elapsed time 0.82
Elapsed time 0.46
(True,
 1,
 3,
 [('fmzfe/file0.txt', 'LOADED', 3, 3, 1, 0, None, None, None, None)])
```

```
[9] df2 = query(f"SELECT * FROM {TABLE}")
    df2
```

```
Elapsed time 1.65
```

	a	b
0	1	4
1	2	5
2	3	6


Worksheet - New Worksheet (1/
cz86276.ap-southeast-1.snowflakecomputing.com/console#/internal/worksheet
Databases
Shares
Data Marketplace
Warehouses
Worksheets
History
Preview App
Partner Connect
Help
JGRAVES
SYSADMIN
New Worksheet
Find database objects
Starting with...
DEMO_DB
INFORMATION_SCHEMA
PUBLIC
Tables
NZPUG_DEMO
TEST
No Views in this Schema
SNOWFLAKE_SAMPLE_DATA
INFORMATION_SCHEMA
TPCDS_SF100TCL
TPCDS_SF10TCL
TPCH_SF001
TPCH_SF1
TPCH_SF10
TPCH_SF100
TPCH_SF1000
TPCH_SF10000
WEATHER
Tables
DAILY_14_TOTAL
DAILY_16_TOTAL
HOURLY_14_TOTAL
Table: NZPUG_DEMO
Created on: 7/20/2021, 10:39:25 PM
Owner: SYSADMIN
Rows: 3
Size: 1.0KB
Run
All Queries
Saved 13 minutes ago
SYSADMIN
COMPUTE_WH (XS)
DEMO_DB
PUBLIC
Results
Data Preview
Open History
Table: DEMO_DB.PUBLIC.NZPUG_DEMO
Data
Details
Filter result...
Columns
Row
a
b
1
1
4
2
2
5
3
3
6

TIPS & TRICKS

REDUCING TIME TO INSIGHT

- **Use dotenv with .env file**

```
from dotenv import load_dotenv
conn = snowflake.connector.connect(user=os.getenv('user'),
                                   password=os.getenv('password'),
                                   …)
```

- **Use triple quotes**

```
df = query("""SELECT * FROM <TABLE>
""")
```

- **Two step query**

```
execute("show columns in table <TABLE>;")
df = query("""select * from table(result_scan(last_query_id()));""")
```

AutoHotkey_Script.ahk - Notepad

File Edit Format View Help

```
#NoEnv  ; Recommended for performance and compatibility with future AutoHotkey releases.
; #Warn  ; Enable warnings to assist with detecting common errors.
SendMode Input  ; Recommended for new scripts due to its superior speed and reliability.
SetWorkingDir %A_ScriptDir%  ; Ensures a consistent starting directory.

; Key:
; # Windows key
; ^ Control key
; + Shift key
; ! Alt key

; CONTROL SHIFT 1: Send string
^+1::Send {Raw}`%run "C:\\setup.py"

; CONTROL SHIFT 3: Send string
^+3::Send .copy`(`)!{enter}

; CONTROL SHIFT 5: Send string
^+5::Send pd.read_feather(){left}

; CONTROL SHIFT 7: Send string
^+7::Send `, sort_cols=True

; CONTROL SHIFT 8: Send string
^+8::Send .reset_index`(inplace=True, drop=True`)

; CONTROL SHIFT F8: Send string
^+F8::Send {left}{Backspace}{Backspace}{Backspace}{Backspace}{Backspace}{Backspace}{Backspace}{Backspace}{Backspace}{Backspace}{Backspace}{Backspace}!{Enter}

; CONTROL SHIFT 9: Send string
^+9::Send _ = [print`(item`) for item in sorted`(list`(df.columns`)`)]{left}{left}{left}{left}{left}{left}{left}{left}{left}{left}{left}{left}

; CONTROL SHIFT [: Send string
^+[::Send [:2]!{enter}
```

AUTOHOTKEY CHEATSHEET

CONTROL SHIFT 0: 02121383

CONTROL SHIFT 1: %run "C:\\ setup.py

CONTROL SHIFT 3: .copy() Alt-Enter

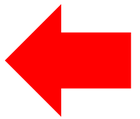
CONTROL SHIFT 5: pd.read_feather() {left}

CONTROL SHIFT 7: , sort_cols=True

CONTROL SHIFT 8: .reset_index(inplace=True, drop=True) (see F8)
CONTROL SHIFT F8: Delete , drop=True

CONTROL SHIFT 9: sorted(list(df.columns))

CONTROL SHIFT [: [:2] Alt-Enter

CONTROL SHIFT]: .stb.freq(['']) 

CONTROL SHIFT \: ['']

CONTROL SHIFT .: Copy “df*x*” to next line

CONTROL SHIFT /: Copy “df*xx*” to next line

CONTROL SHIFT A: .rename(columns={'count':'_count'})

CONTROL SHIFT D: Find next df number

CONTROL SHIFT E: john.graves@

CONTROL SHIFT F: Add path/filename formula to Excel

CONTROL SHIFT G: Set up for bar chart with title on clipboard

CONTROL SHIFT I: inplace=True

CONTROL SHIFT K: .pivot(index='' , columns='' , values='')

CONTROL SHIFT L: .apply(lambda row: row., axis=1)

CONTROL SHIFT N: "NOW" types date as YYYYMMDD HHMM

CONTROL SHIFT Q: query("""SELECT * FROM """)

CONTROL SHIFT R: "NOW" types date as YYYYMMDD HHMM - replacing existing

CONTROL SHIFT T: "TODAY" types date as YYYYMMDD

CONTROL SHIFT X: .to_csv("", index=False)

CONTROL SHIFT Z: e(start_time) Ctrl+Enter

LINKS

WHERE TO LEARN MORE

- **Github for this talk**
  https://bit.ly/nzpug210721
- **Snowflake with Free Trial**

README.md

nzpug20210721

New Zealand Python User Group Auckland 2021-07-21

Links

Snowflake vs Athena

Literate programming - Wikipedia

Project Jupyter - Wikipedia (Try Free) Architecture — Jupyter Documentation 4.1.1 alpha documentation

Python (programming language) - Wikipedia

pandas (software) - Wikipedia

sidetable

Plotly - Wikipedia

AutoHotkey (for Windows)

How to be more productive as a Data Scientist

Suggested Usage

- Clone this repository on your local machine where you have the Anaconda Python Distribution installed with Jupyter