Python Libraries Guide with Examples Data Science & Math Libraries numpy ●​ Makes working with numbers and arrays super fast ●​ Think Excel but for millions of numbers at once ●​ Essential for any data work - like the foundation of a house Example: python import numpy as np # Create array and do math on millions of numbers instantly arr = np.array([1, 2, 3, 4, 5]) result = np.sqrt(arr) \* 2 # [2.0, 2.83, 3.46, 4.0, 4.47] pandas ●​ Works with data tables (like Excel spreadsheets) ●​ Read CSV files, clean messy data, filter and sort information ●​ Your best friend for data analysis Example: python import pandas as pd # Read Excel/CSV file and analyze data df = pd.read\_csv('sales\_data.csv') monthly\_sales = df.groupby('month')['sales'].sum() top\_products = df.nlargest(5, 'revenue') matplotlib ●​ Creates charts and graphs from your data ●​ Line charts, bar charts, pie charts - you name it ●​ Basic but powerful plotting tool Example: python import matplotlib.pyplot as plt # Create a simple line chart

================================================== Page 2 ==================================================

x = [1, 2, 3, 4, 5] y = [2, 5, 3, 8, 7] plt.plot(x, y) plt.title('Sales Over Time') plt.show() seaborn ●​ Makes matplotlib charts look beautiful and professional ●​ Creates statistical plots with just one line of code ●​ Like Instagram filters but for data charts Example: python import seaborn as sns # Beautiful correlation heatmap in one line sns.heatmap(df.corr(), annot=True, cmap='coolwarm') plt.show() scipy ●​ Advanced math functions for science and engineering ●​ Calculus, statistics, optimization problems ●​ When regular math isn't enough Example: python from scipy import stats from scipy.optimize import minimize # Statistical tests and optimization t\_stat, p\_value = stats.ttest\_ind(group1, group2) result = minimize(cost\_function, initial\_guess) sympy ●​ Does algebra and calculus symbolically (like on paper) ●​ Solves equations, derivatives, integrals automatically ●​ Your digital math tutor Example: python import sympy as sp x = sp.Symbol('x') # Solve equation: x^2 + 2x - 3 = 0

================================================== Page 3 ==================================================

equation = x\*\*2 + 2\*x - 3 solution = sp.solve(equation, x) # [-3, 1] Machine Learning & AI scikit-learn ●​ The Swiss Army knife of machine learning ●​ Classification, regression, clustering made simple ●​ Perfect for beginners and experts alike Example: python from sklearn.ensemble import RandomForestClassifier from sklearn.model\_selection import train\_test\_split # Train a model to predict categories X\_train, X\_test, y\_train, y\_test = train\_test\_split(data, labels) model = RandomForestClassifier() model.fit(X\_train, y\_train) predictions = model.predict(X\_test) tensorflow ●​ Google's powerful deep learning framework ●​ Build neural networks and AI models ●​ Used by major companies worldwide Example: python import tensorflow as tf # Build a neural network model = tf.keras.Sequential([ tf.keras.layers.Dense(128, activation='relu'), tf.keras.layers.Dense(10, activation='softmax') ]) model.compile(optimizer='adam', loss='categorical\_crossentropy') keras ●​ Makes TensorFlow easier to use ●​ Build deep learning models with simple code ●​ Like having training wheels for AI

================================================== Page 4 ==================================================

Example: python from keras.models import Sequential from keras.layers import Dense # Simple neural network in 3 lines model = Sequential() model.add(Dense(64, activation='relu', input\_shape=(784,))) model.add(Dense(10, activation='softmax')) torch (PyTorch) ●​ Facebook's deep learning framework ●​ More flexible than TensorFlow for research ●​ Preferred by many AI researchers Example: python import torch import torch.nn as nn # Create a neural network class SimpleNet(nn.Module): def \_\_init\_\_(self): super().\_\_init\_\_() self.fc1 = nn.Linear(784, 128) self.fc2 = nn.Linear(128, 10) xgboost ●​ Super powerful machine learning algorithm ●​ Wins many data science competitions ●​ Great for structured/tabular data Example: python import xgboost as xgb # Train XGBoost model model = xgb.XGBClassifier() model.fit(X\_train, y\_train) predictions = model.predict(X\_test) # Often gets 95%+ accuracy! transformers

================================================== Page 5 ==================================================

●​ Access to state-of-the-art AI language models ●​ BERT, GPT, RoBERTa and more ●​ Hugging Face's amazing model hub Example: python from transformers import pipeline # Sentiment analysis in 2 lines classifier = pipeline("sentiment-analysis") result = classifier("I love this movie!") # POSITIVE: 0.99 Text & Language Processing nltk ●​ The grandfather of Python NLP libraries ●​ Tokenize text, remove stopwords, analyze grammar ●​ Academic-focused natural language toolkit Example: python import nltk from nltk.tokenize import word\_tokenize from nltk.corpus import stopwords # Clean and analyze text text = "The quick brown fox jumps" tokens = word\_tokenize(text) clean\_tokens = [w for w in tokens if w not in stopwords.words('english')] spacy ●​ Industrial-strength text processing ●​ Fast entity recognition, POS tagging, parsing ●​ Used in production by real companies Example: python import spacy nlp = spacy.load("en\_core\_web\_sm") # Extract people, places, organizations doc = nlp("Apple Inc. is located in Cupertino, California") for ent in doc.ents:

================================================== Page 6 ==================================================

print(ent.text, ent.label\_) # Apple Inc. ORG, Cupertino GPE textblob ●​ Simple sentiment analysis and text processing ●​ Beginner-friendly NLP operations ●​ Quick and dirty text analysis Example: python from textblob import TextBlob # Sentiment analysis in one line blob = TextBlob("I love this product!") print(blob.sentiment.polarity) # 0.5 (positive) openai ●​ Official way to use OpenAI's APIs ●​ Connect to GPT models, DALL-E, Whisper ●​ Your gateway to cutting-edge AI Example: python import openai # Chat with GPT response = openai.ChatCompletion.create( model="gpt-3.5-turbo", messages=[{"role": "user", "content": "Explain AI in simple terms"}] ) Web Development flask ●​ Lightweight web framework ●​ Build web apps and APIs quickly ●​ Minimal and flexible - you choose what to add Example: python from flask import Flask app = Flask(\_\_name\_\_)

================================================== Page 7 ==================================================

@app.route('/') def hello(): return "Hello World!" app.run() # Website running in 5 lines! django ●​ Full-featured web framework with everything included ●​ User authentication, admin panel, database ORM ●​ The "batteries included" web framework Example: python # In views.py from django.shortcuts import render from django.http import HttpResponse def home(request): return HttpResponse("Welcome to my site!") # Comes with user login, admin panel, database built-in fastapi ●​ Modern, super-fast API framework ●​ Automatic API documentation generation ●​ Type hints make development smoother Example: python from fastapi import FastAPI app = FastAPI() @app.get("/users/{user\_id}") def get\_user(user\_id: int): return {"user\_id": user\_id, "name": "John"} # Automatic API docs at /docs streamlit ●​ Turn data scripts into web apps instantly ●​ No HTML/CSS needed - just Python

================================================== Page 8 ==================================================

●​ Perfect for data science demos Example: python import streamlit as st import pandas as pd # Create web app in 3 lines st.title("My Data App") df = pd.read\_csv('data.csv') st.dataframe(df) # Interactive table on web! requests ●​ Makes HTTP requests simple and human-friendly ●​ Call APIs, scrape websites, download files ●​ Every Python developer uses this Example: python import requests # Get data from any API response = requests.get('https://api.github.com/users/octocat') user\_data = response.json() print(user\_data['name']) # The Octocat beautifulsoup4 ●​ Parse and extract data from HTML/XML ●​ Web scraping made easy ●​ Navigate webpage structure like a tree Example: python from bs4 import BeautifulSoup import requests # Scrape website data page = requests.get('https://quotes.toscrape.com') soup = BeautifulSoup(page.content, 'html.parser') quotes = soup.find\_all('span', class\_='text') Interactive Data Visualization plotly

================================================== Page 9 ==================================================

●​ Create interactive charts for web browsers ●​ Zoom, hover, click on your visualizations ●​ Professional-looking dashboards Example: python import plotly.express as px # Interactive chart in one line fig = px.scatter(df, x='height', y='weight', color='gender') fig.show() # Hover, zoom, pan automatically! dash ●​ Build analytical web applications with Python ●​ No JavaScript needed for interactive apps ●​ Plotly's web app framework Example: python import dash from dash import dcc, html # Interactive dashboard app = dash.Dash(\_\_name\_\_) app.layout = html.Div([ dcc.Graph(figure=px.bar(df, x='month', y='sales')) ]) wordcloud ●​ Generate word clouds from text ●​ Visualize word frequency in creative ways ●​ Great for text analysis presentations Example: python from wordcloud import WordCloud # Create word cloud from text text = "python data science machine learning" wordcloud = WordCloud().generate(text) plt.imshow(wordcloud) plt.show() Computer Vision & Images

================================================== Page 10 ==================================================

opencv-python ●​ Computer vision and image processing powerhouse ●​ Face detection, object tracking, image filters ●​ Used in robotics, security, and automation Example: python import cv2 # Face detection in 3 lines face\_cascade = cv2.CascadeClassifier('haarcascade\_frontalface\_default.xml') img = cv2.imread('photo.jpg') faces = face\_cascade.detectMultiScale(img, 1.1, 4) pillow (PIL) ●​ Basic image operations made simple ●​ Resize, crop, rotate, convert images ●​ Essential for any image manipulation Example: python from PIL import Image # Resize image and add filter img = Image.open('photo.jpg') img\_resized = img.resize((800, 600)) img\_resized.save('photo\_small.jpg') Database & Storage pymongo ●​ Connect Python to MongoDB databases ●​ Work with NoSQL document databases ●​ Great for flexible, schema-less data Example: python from pymongo import MongoClient client = MongoClient('mongodb://localhost:27017/') db = client['my\_database'] collection = db['users'] # Insert and find documents

================================================== Page 11 ==================================================

collection.insert\_one({'name': 'John', 'age': 30}) sqlalchemy ●​ Object-Relational Mapping (ORM) for SQL databases ●​ Write Python code instead of SQL queries ●​ Works with PostgreSQL, MySQL, SQLite, etc. Example: python from sqlalchemy import create\_engine, Column, Integer, String from sqlalchemy.ext.declarative import declarative\_base Base = declarative\_base() class User(Base): \_\_tablename\_\_ = 'users' id = Column(Integer, primary\_key=True) name = Column(String) # Python objects = Database tables redis ●​ In-memory data structure store ●​ Caching, session storage, message queuing ●​ Super fast key-value database Example: python import redis r = redis.Redis(host='localhost', port=6379) # Lightning fast data storage r.set('user:1000', 'John Doe') name = r.get('user:1000') # Retrieved in microseconds Communication & Messaging twilio ●​ Send SMS, make calls, video chat programmatically ●​ Communication APIs made simple ●​ Build messaging and calling features Example:

================================================== Page 12 ==================================================

python from twilio.rest import Client client = Client(account\_sid, auth\_token) # Send SMS in 3 lines message = client.messages.create( body="Hello from Python!", from\_='+1234567890', to='+0987654321' ) yagmail ●​ Send emails with minimal code ●​ Simplified email sending ●​ No complex SMTP configuration needed Example: python import yagmail # Send email in 2 lines yag = yagmail.SMTP('your@gmail.com', 'password') yag.send('recipient@email.com', 'Subject', 'Email body') tweepy ●​ Twitter API wrapper for Python ●​ Tweet, read timelines, analyze Twitter data ●​ Social media automation and analysis Example: python import tweepy # Tweet automatically auth = tweepy.OAuthHandler(consumer\_key, consumer\_secret) api = tweepy.API(auth) api.update\_status("Hello Twitter from Python!") Jupyter & Interactive Computing ipywidgets ●​ Interactive widgets for Jupyter notebooks ●​ Sliders, buttons, dropdowns in notebooks

================================================== Page 13 ==================================================

●​ Make notebooks interactive without coding Example: python import ipywidgets as widgets from IPython.display import display # Interactive slider slider = widgets.IntSlider(value=7, min=0, max=10) display(slider) # Changes update in real-time! voila ●​ Turn notebooks into standalone web apps ●​ Share interactive dashboards easily ●​ No server management required Example: bash # Convert notebook to web app voila mynotebook.ipynb # Now it's a web application! Data Analysis & Profiling pandas-profiling ●​ Generate comprehensive data reports automatically ●​ Understand your dataset without writing code ●​ One-line exploratory data analysis Example: python from pandas\_profiling import ProfileReport import pandas as pd # Complete data analysis in one line df = pd.read\_csv('data.csv') profile = ProfileReport(df, title="Data Analysis Report") profile.to\_file("report.html") sweetviz ●​ Beautiful automated EDA reports

================================================== Page 14 ==================================================

●​ Compare datasets visually ●​ Stunning data analysis visualizations Example: python import sweetviz as sv # Beautiful EDA report report = sv.analyze(df) report.show\_html("sweet\_report.html") missingno ●​ Visualize missing data patterns ●​ Understand where your data is incomplete ●​ Essential for data cleaning Example: python import missingno as msno # Visualize missing data patterns msno.matrix(df) # Shows missing data heatmap msno.bar(df) # Bar chart of missing values Model Explanation & Fairness shap ●​ Unified framework for model explanations ●​ Game theory approach to feature importance ●​ Industry standard for model interpretability Example: python import shap # Explain any machine learning model explainer = shap.TreeExplainer(model) shap\_values = explainer.shap\_values(X\_test) shap.summary\_plot(shap\_values, X\_test) lime ●​ Local explanations for individual predictions ●​ Understand why a model predicted X for this case

================================================== Page 15 ==================================================

●​ Black-box model explanation Example: python from lime.lime\_text import LimeTextExplainer # Explain individual predictions explainer = LimeTextExplainer(class\_names=['negative', 'positive']) exp = explainer.explain\_instance(text\_instance, classifier\_fn) Maps & Geographic Data folium ●​ Create interactive maps in Python ●​ Leaflet.js maps with Python data ●​ Beautiful web maps with minimal code Example: python import folium # Interactive map in 3 lines m = folium.Map(location=[45.5236, -122.6750]) folium.Marker([45.5236, -122.6750], popup='Portland').add\_to(m) m.save('map.html') geopandas ●​ Pandas but for geographic data ●​ Work with shapefiles, GeoJSON, coordinates ●​ Spatial data analysis made simple Example: python import geopandas as gpd # Load and analyze geographic data world = gpd.read\_file(gpd.datasets.get\_path('naturalearth\_lowres')) world.plot() # World map in one line! Game Development & Desktop Apps pygame ●​ Create games and multimedia applications

================================================== Page 16 ==================================================

●​ 2D graphics, sound, input handling ●​ Popular for indie game development Example: python import pygame pygame.init() screen = pygame.display.set\_mode((800, 600)) # Game loop running = True while running: for event in pygame.event.get(): if event.type == pygame.QUIT: running = False tkinter ●​ Built-in GUI toolkit ●​ Simple desktop applications ●​ Comes with Python, no installation needed Example: python import tkinter as tk # Desktop app in 4 lines root = tk.Tk() root.title("My App") tk.Label(root, text="Hello World!").pack() root.mainloop() kivy ●​ Cross-platform GUI development ●​ Mobile apps, desktop apps, touch interfaces ●​ Multi-touch application framework Example: python from kivy.app import App from kivy.uix.label import Label class MyApp(App): def build(self): return Label(text='Hello Kivy!')

================================================== Page 17 ==================================================

MyApp().run() Testing & Code Quality pytest ●​ Modern testing framework for Python ●​ Write simple tests with powerful features ●​ Most popular Python testing tool Example: python # test\_math.py def add(a, b): return a + b def test\_add(): assert add(2, 3) == 5 assert add(-1, 1) == 0 # Run: pytest test\_math.py selenium ●​ Automate web browsers for testing ●​ Click buttons, fill forms, scrape data ●​ Most popular web automation framework Example: python from selenium import webdriver # Automate browser driver = webdriver.Chrome() driver.get("https://google.com") search\_box = driver.find\_element("name", "q") search\_box.send\_keys("Python") search\_box.submit() Background Tasks & Scheduling celery

================================================== Page 18 ==================================================

●​ Distributed task queue ●​ Background job processing ●​ Scale your application with workers Example: python from celery import Celery app = Celery('tasks', broker='redis://localhost:6379') @app.task def send\_email(email, message): # Send email in background return "Email sent!" # Run: send\_email.delay('user@email.com', 'Hello') schedule ●​ Human-friendly task scheduling ●​ Simple syntax for recurring jobs ●​ Lightweight job scheduler Example: python import schedule import time def backup\_data(): print("Backing up data...") # Schedule tasks in plain English schedule.every(10).minutes.do(backup\_data) schedule.every().hour.do(backup\_data) schedule.every().day.at("10:30").do(backup\_data) Date & Time Handling arrow ●​ Better dates and times for Python ●​ Human-friendly date manipulation ●​ Timezone-aware date operations Example:

================================================== Page 19 ==================================================

python import arrow # Human-friendly dates now = arrow.now() tomorrow = now.shift(days=+1) print(now.humanize()) # "just now" print(tomorrow.format('YYYY-MM-DD')) dateutil ●​ Powerful extensions to Python's datetime ●​ Parse dates in any format ●​ Complex date arithmetic and parsing Example: python from dateutil import parser from dateutil.relativedelta import relativedelta # Parse any date format date = parser.parse("March 14, 2023") next\_month = date + relativedelta(months=1) Fun Utilities & Automation qrcode ●​ Generate QR codes ●​ Create QR codes for URLs, text, data ●​ Customizable QR code generation Example: python import qrcode # Generate QR code qr = qrcode.QRCode(version=1, box\_size=10, border=5) qr.add\_data('https://www.python.org') qr.make(fit=True) img = qr.make\_image(fill\_color="black", back\_color="white") faker ●​ Generate fake data for testing ●​ Names, addresses, emails, phone numbers

================================================== Page 20 ==================================================

●​ Mock data for development Example: python from faker import Faker fake = Faker() # Generate fake data for testing print(fake.name()) # "John Smith" print(fake.email()) # "john@example.com" print(fake.address()) # "123 Main St, City, State" pyautogui ●​ Programmatic control of mouse and keyboard ●​ Automate GUI interactions ●​ Screen automation and testing Example: python import pyautogui # Automate mouse and keyboard pyautogui.click(100, 200) # Click at coordinates pyautogui.typewrite('Hello World') # Type text pyautogui.press('enter') # Press key Terminal Enhancement & Display rich ●​ Rich text and beautiful formatting ●​ Colors, tables, progress bars, syntax highlighting ●​ Make your terminal output beautiful Example: python from rich.console import Console from rich.table import Table console = Console() table = Table(show\_header=True, header\_style="bold magenta") table.add\_column("Name", style="dim", width=12) table.add\_column("Age") table.add\_row("John", "25")

================================================== Page 21 ==================================================

console.print(table) tqdm ●​ Fast, extensible progress bars ●​ Track progress of long operations ●​ Most popular Python progress bar Example: python from tqdm import tqdm import time # Beautiful progress bar for i in tqdm(range(100)): time.sleep(0.01) # Your long operation # 76%|████████▎ | 76/100 [00:07<00:02, 9.96it/s] Audio Processing pydub ●​ Audio manipulation with simple interface ●​ Cut, concatenate, convert audio files ●​ Audio processing made easy Example: python from pydub import AudioSegment # Load and edit audio song = AudioSegment.from\_wav("song.wav") first\_10\_seconds = song[:10000] # First 10 seconds louder\_song = song + 6 # Increase volume by 6dB pyttsx3 ●​ Text-to-speech conversion ●​ Make your computer speak text ●​ Cross-platform speech synthesis Example: python import pyttsx3 # Make computer speak

================================================== Page 22 ==================================================

engine = pyttsx3.init() engine.say("Hello, I am speaking from Python!") engine.runAndWait() System & Hardware Interaction keyboard ●​ Global hotkeys and keyboard events ●​ Detect key presses system-wide ●​ Keyboard automation and monitoring Example: python import keyboard # Detect keyboard events def on\_key\_press(event): print(f'Key {event.name} pressed') keyboard.on\_press(on\_key\_press) keyboard.wait('esc') # Wait until ESC is pressed watchdog ●​ File system event monitoring ●​ Watch for file changes automatically ●​ Real-time file system notifications Example: python from watchdog.observers import Observer from watchdog.events import FileSystemEventHandler class MyHandler(FileSystemEventHandler): def on\_modified(self, event): print(f'File {event.src\_path} was modified') observer = Observer() observer.schedule(MyHandler(), path='.') observer.start() Built-in Python Libraries (Standard Library)

================================================== Page 23 ==================================================

json ●​ JSON encoder and decoder ●​ Work with JSON data format ●​ API communication and data storage Example: python import json # Work with JSON data data = {'name': 'John', 'age': 30} json\_string = json.dumps(data) # Convert to JSON parsed\_data = json.loads(json\_string) # Parse JSON csv ●​ CSV file reading and writing ●​ Handle comma-separated value files ●​ Spreadsheet data processing Example: python import csv # Read CSV file with open('data.csv', 'r') as file: reader = csv.reader(file) for row in reader: print(row) # ['Name', 'Age', 'City'] re ●​ Regular expression operations ●​ Pattern matching in strings ●​ Text parsing and validation Example: python import re # Find patterns in text text = "My phone number is 123-456-7890" pattern = r'\d{3}-\d{3}-\d{4}' phone = re.search(pattern, text).group() # "123-456-7890" pathlib

================================================== Page 24 ==================================================

●​ Object-oriented filesystem paths ●​ Modern way to handle file paths ●​ Better than os.path for file operations Example: python from pathlib import Path # Modern file path handling path = Path('data/files/document.txt') print(path.parent) # data/files print(path.suffix) # .txt path.mkdir(parents=True, exist\_ok=True) # Create directories collections ●​ Specialized container datatypes ●​ Counter, defaultdict, OrderedDict ●​ Enhanced data structures Example: python from collections import Counter, defaultdict # Count occurrences words = ['apple', 'banana', 'apple', 'cherry'] count = Counter(words) # Counter({'apple': 2, 'banana': 1, 'cherry': 1}) # Dictionary with default values dd = defaultdict(list) dd['fruits'].append('apple') # No KeyError!