

Technical Skill Cluster Workshop #1: File Paths

2026-02-3

1. Path Syntax by Operating System

| System | Separator | Example Path (Absolute) |
|-------------|-----------------|-------------------------|
| Windows | Backslash \ | C:\Users\name\data.csv |
| Mac / Linux | Forward slash / | /Users/name/data.csv |

Fun fact: Both Python and R accept forward slashes / on ALL operating systems. Use forward slashes and your code will work everywhere!

2. How to Copy a File Path

Windows:

- Hold Shift + Right-click the file → “Copy as path”
- Or: Click the address bar in File Explorer and copy

Mac:

- Right-click the file → Hold Option → “Copy as Pathname”
- Or: Drag file into Terminal to see path

3. Python (with pandas)

Check your working directory

```
import os
print(os.getcwd())    # Shows current folder
```

Import data

```
import pandas as pd

# Option 1: Forward slashes (recommended, works everywhere)
df = pd.read_csv("C:/Users/name/Documents/data.csv")

# Option 2: Raw string with backslashes (Windows only)
df = pd.read_csv(r"C:\Users\name\Documents\data.csv")

# Option 3: Relative path (file in working directory)
df = pd.read_csv("data.csv")
```

View your data

```
df.head()           # First 5 rows
df.shape            # (rows, columns)
df.columns          # Column names
```

4. R

Check your working directory

```
getwd()             # Shows current folder
setwd("C:/path")    # Change working directory
```

Import data

```
# Option 1: Forward slashes (recommended, works everywhere)
df <- read.csv("C:/Users/name/Documents/data.csv")

# Option 2: Relative path (file in working directory)
df <- read.csv("data.csv")
```

View your data

```
head(df)      # First 6 rows
dim(df)       # Rows and columns
names(df)     # Column names
View(df)      # Open in spreadsheet viewer (RStudio)
```

5. Common Mistakes to Avoid

| Mistake | Problem | Fix |
|-----------------------|--|---|
| Forgetting quotes | <code>read.csv(data.csv)</code> | Add quotes: <code>"data.csv"</code> |
| Backslashes in Python | <code>\U</code> and <code>\n</code> are escape codes | Use <code>/</code> or raw string <code>r"..."</code> |
| File not found | Wrong working directory | Check with <code>getwd()</code> or <code>os.getcwd()</code> |
| Typos in path | Misspelled folder name | Copy path directly from file explorer |

6. Absolute vs. Relative Paths

| Type | Example | When to Use |
|----------|---|----------------------------------|
| Absolute | <code>C:/Users/name/project/data.csv</code> | One-off scripts, quick tasks |
| Relative | <code>./data.csv</code> | Shared projects, reproducibility |

Tip: Relative paths make your code portable. If you share your project folder, others can run it without changing paths.

Dot Notation

| Symbol | Meaning | Example |
|-----------------|--|--------------------------|
| <code>.</code> | Current directory (where your script is running) | <code>./data.csv</code> |
| <code>..</code> | Parent directory (one folder up) | <code>../data.csv</code> |

You can chain `..` to go up multiple levels: `../../file.csv` goes up two folders.

Example: Navigating a Project Folder

Imagine your project has this structure:

```
my_project/
+-- code/
|   +-- analysis.py
|   +-- helpers/
|   +-- utils.py
+-- data/
|   +-- raw/
|   |   +-- survey.csv
|   +-- cleaned/
|       +-- survey_clean.csv
```

```

+-- output/
+-- results.csv

```

If your **working directory** is `my_project/code/`, here's how to access different files:

| Target File | Relative Path | Explanation |
|------------------|----------------------------------|---|
| survey.csv | ../data/raw/survey.csv | Go up to my_project/, then into data/raw/ |
| survey_clean.csv | ../data/cleaned/survey_clean.csv | Go up to my_project/, then into data/cleaned/ |
| results.csv | ../output/results.csv | Go up to my_project/, then into output/ |

If your **working directory** is `my_project/code/helpers/`, accessing `survey.csv`:

```

../../data/raw/survey.csv
|  |
|  +-- First `..` goes from helpers/ up to code/
+----- Second `..` goes from code/ up to my_project/

```

Python & R Examples

```

# Python - from code/ folder, read raw data
df = pd.read_csv("../data/raw/survey.csv")

# Save cleaned data
df.to_csv("../data/cleaned/survey_clean.csv", index=False)

# R - from code/ folder, read raw data
df <- read.csv("../data/raw/survey.csv")

# Save cleaned data
write.csv(df, "../data/cleaned/survey_clean.csv", row.names = FALSE)

```

Quick Reference

| Task | Python | R |
|--------------------------|--------------------------------------|-----------------------------------|
| Check working directory | <code>os.getcwd()</code> | <code>getwd()</code> |
| Change working directory | <code>os.chdir("path")</code> | <code>setwd("path")</code> |
| Read CSV | <code>pd.read_csv("file.csv")</code> | <code>read.csv("file.csv")</code> |
| View first rows | <code>df.head()</code> | <code>head(df)</code> |
| View dimensions | <code>df.shape</code> | <code>dim(df)</code> |