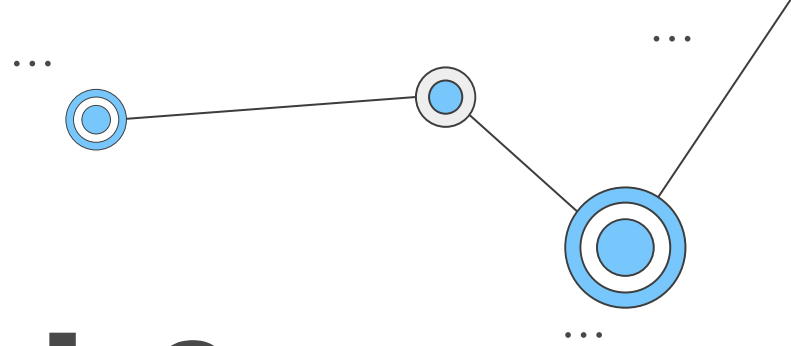




**MAISI**



# **CRA Week 6:**

## **Kaplan–Meier Curves and Survival Analysis cont.**

Michigan Data Science Team  
Fall 2025

# Session 6 Agenda

01

## Fun Icebreaker!!

Get to know your projectmates!

...

02

## Survival Analysis Recap

Let's go over what we learned last week.

...

03

## Kaplan-Meier Curves

What is this survival model?

...

04

## Interpreting Kaplan-Meier Curves

What do our findings tell us about our dataset?

...

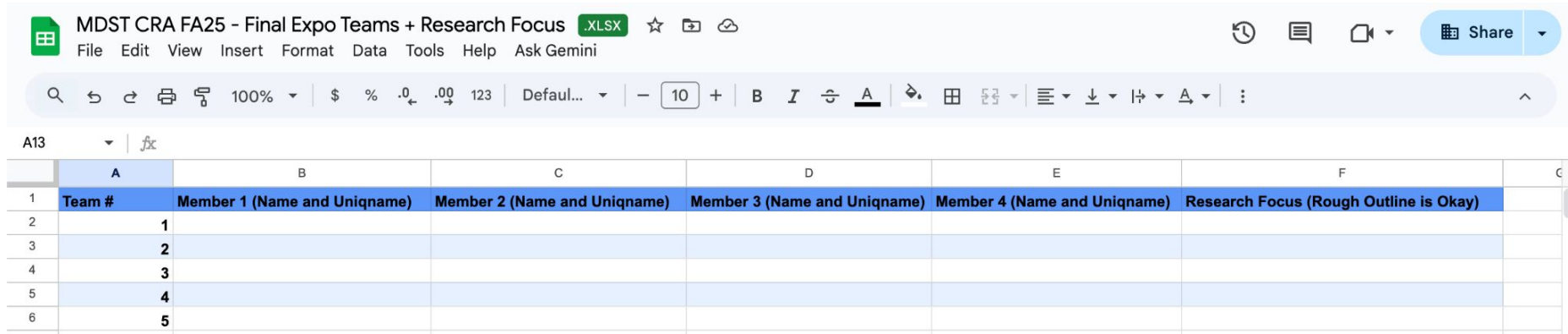
05

## Practice Time!

Work on the KM notebook!

...

# But first...



The screenshot shows a Google Sheets interface. The title bar reads 'MDST CRA FA25 - Final Expo Teams + Research Focus' with an '.XLSX' extension. The menu bar includes 'File', 'Edit', 'View', 'Insert', 'Format', 'Data', 'Tools', 'Help', and 'Ask Gemini'. The toolbar shows various editing tools, a zoom level of 100%, and a font size of 10. The spreadsheet has columns A through G and rows 1 through 6. The first row is a header with blue background and white text. The subsequent rows have light blue backgrounds.

	A	B	C	D	E	F	G
1	Team #	Member 1 (Name and Uniqname)	Member 2 (Name and Uniqname)	Member 3 (Name and Uniqname)	Member 4 (Name and Uniqname)	Research Focus (Rough Outline is Okay)	
2	1						
3	2						
4	3						
5	4						
6	5						

Talk to the people around you and start thinking about what you will want to present on at the Final Expo! We will go more in-depth as to what kind of topics you might want to look into, but if you have ideas now, feel free to discuss them and sign up! (you can change your mind later)

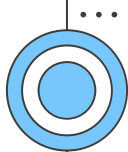
# Quick Icebreaker!!

Share with the people around you,  
and **make up your own** if you're  
feeling creative :)

What are your thoughts on the following food classifications?

- Is a ravioli a dumpling?
- Is an apple a low-entropy salad? (look up 'salad theory')
- Is a hotdog a sandwich?
- Is cereal a soup?





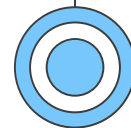
# Survival Analysis Review

- Used for analyzing “**time-to-event**” data (in our case, recidivism)
- Traditionally used in healthcare: “What variables affect the risk of a patient in care dying, and how does this look across time?”
- Example in context:
  - Helps us answer questions like: How long does a person go without reoffending after release?
- Identify factors that influence how long individuals remain “event-free” (without ... reoffending) after release



## Facets of Survival Analysis:

- **Survival Function  $S(t)$ :** Probability of “surviving” (no event) up to time  $t$
- **Hazard Function  $h(t)$ :** Rate of event occurrence at time  $t$



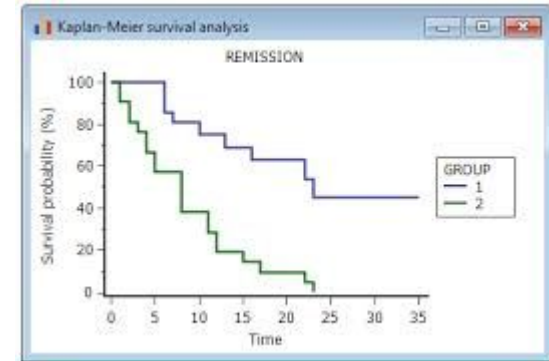
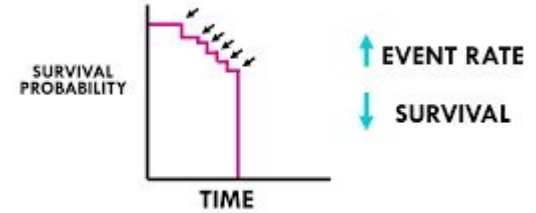
# How does this relate to COMPAS?

- COMPAS predicts the likelihood of **recidivism**
- Survival analysis can help understand if COMPAS predictions are **biased** or **accurate** over **time**
- **Cox Proportional Hazards Model (Cox PH) (Last Week)**
  - Understand which factors (age, race, prior offenses) increase or decrease recidivism risk, and how that interacts with COMPAS scores
- **Kaplan-Meier Curve (Today!)**
  - Understand descriptive trends by demographics (e.g., age, race)
  - How does recidivism probability change over time for different demographic groups?
  - How does COMPAS assign different “meanings” of scores to different groups?

...

# Kaplan-Meier Curves

- **Non-parametric** model used to estimate the **probability** of “**surviving**” (remaining event-free) over time without assuming any specific hazard distribution
- In context, it can help us see how **long** individuals remain **recidivism-free** after release
- Example:
  - Tracking two groups after release—those with prior convictions and those without
  - KM curve shows how many in each group remain event-free (without reoffending) over time
  - Allows us to compare survival between these groups



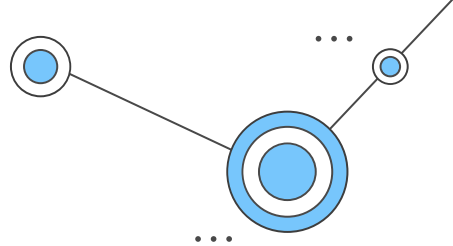
# KM Function

$$S(t) = \prod_{t_i \leq t} \left( 1 - \frac{d_i}{n_i} \right)$$

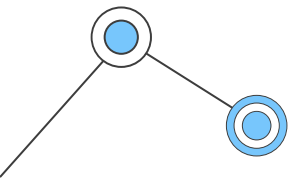
- Big pi: product over all times  $t_i$  up to time  $t$ 
  - $t$ : The time at which we want to know the survival probability (e.g., probability of being event-free at 6 months)
  - $t_i$ : Specific times when events (recidivism) occur within the dataset
- $d_i$  = number of events (e.g., recidivism) occurring at time  $t_i$
- $n_i$  = number of individuals at risk just before  $t_i$
- At each event time  $t_i$  the survival probability is **updated** based on the proportion of individuals who remain event-free (KM curve **steps down** with each observed event)
- Suppose we start with 10 people, and 2 reoffend at  $t = 6$  months
  - The survival probability at 6 months  $1 - \frac{2}{10} = 0.8$
  - If another event happens at 12 months, with 8 people  $S(12) = 0.8 \times (1 - \frac{1}{8}) \approx 0.7$



# Interpreting Kaplan-Meier Curves



- Reading the Curve: Each “**step down**” on the curve represents an **event occurrence** or the probability of **not reoffending** by that **time**
  - Lower curves indicate faster recidivism rates
- Survival Probability at a Given Time: The y-axis value shows the probability of an individual remaining **event-free** up to that point
- Group Comparison: We can plot KM curves for different groups (e.g., individuals with vs. without prior conviction) to visually compare survival rates
- Example:
  - If the curve for those with prior offenses drops more **steeply**, it suggests they are **more likely** to reoffend earlier than those without prior offenses
  - If the survival probability for a group is 0.5 at 1 year, it means that 50% of the group is expected to remain event-free for at least one year



# Calculating and Interpreting Median Survival Times

- Overall **median survival** provides a **benchmark** for when 50% of the population reoffends
- Compare survival times for **Low**, **Medium**, and **High** risk to understand how risk scores correlate with recidivism timing

# Use subset['end'] as the duration data, subset['two\_year\_recid'] as event the indicator

```
kmf.fit(durations=subset['end'], event_observed=subset['two_year_recid'])
```

# median\_survival\_time\_ is an attribute of the fitted Kaplan-Meier model

```
median_survival = kmf.median_survival_time_
```

```
print("Median Survival:", median_survival)
```

- If **median\_survival\_time\_** is a number (e.g., 400 days), it means that **half** of the individuals in this group recidivated within 400 days of release.
- If **median\_survival\_time\_** is **inf** (infinity), it means that more than 50% of the group **didn't** experience recidivism within the observed time frame. In other words, the median survival time **exceeds** the maximum observation period.

# Comparing KM using Python

- Analyze survival curves for **Low**, **Medium**, and **High** risk groups to see differences in recidivism
- For example:

**for risk, color in zip(risk\_levels, colors):**

```
# Isolate data for each specific risk level
```

```
subset = df[df['score_text'] == risk]
```

```
# Use subset['end'] as the duration data, subset['two_year_recid'] as event the indicator
```

```
kmf.fit(durations=subset['end'], event_observed=subset['two_year_recid'])
```

```
kmf.plot(label=f'Risk Level: {risk}') # Plots, sets labels
```

- **zip()** takes two or more sequences and pairs their elements together into tuples, creating a new sequence of tuples
  - Suppose you have two lists, `risk_levels = ['Low', 'Medium', 'High']` and `colors = ['blue', 'orange', 'green']`
  - When you use `zip(risk_levels, colors)`, it combines these lists into pairs: `[('Low', 'blue'), ('Medium', 'orange'), ('High', 'green')]` ...

# Cox PH vs. Kaplan-Meier Comparison

## Cox PH:

- **Inferential** Analysis: Assesses the effect of **multiple covariates** on the **hazard rate** (risk of event occurring)
- Provides hazard ratios, quantifying how each **factor influences** the risk of reoffending
- Controls for several covariates at once

Cox PH gives **statistical evidence** on covariate effects, KM provides a clear visual summary of **survival trends**

...

Ex. KM shows if individuals with prior offenses reoffend sooner, while Cox PH confirms and quantifies the effect of prior offense on recidivism risk

## KM Curves:

- **Descriptive** Analysis: Shows the **probability** of “**surviving**” (remaining event-free) over **time** for groups
- Compare survival probabilities across subgroups **without** considering additional covariates

# Real World Significance

**Higher** risk levels have **shorter survival** times, indicating **earlier recidivism**

**Variances** in curves for different genders and races highlight potential systemic **disparities**

Key question: **Even though different groups may have different base risk rates, do COMPAS risk scores have the same significance across groups?**

**Median survival** times and curves help prioritize **interventions** and indicate areas for potential **model adjustments**

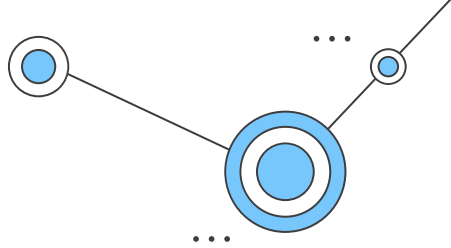
Consider how recidivism data can improve **support services** and ensure **fair, data-driven decision-making**

A decorative network diagram consisting of several blue circular nodes of varying sizes connected by thin grey lines. The nodes are arranged in a non-linear fashion, with some having multiple connections. The largest node is located in the upper right quadrant, and another large node is in the lower left. Smaller nodes are scattered throughout, connected by lines that form a web-like structure. Some nodes have concentric circles, giving them a target-like appearance.

# Practice Time!

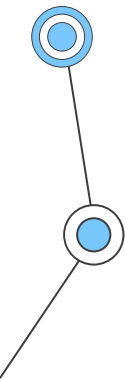
Let's check out the survival rates across groups  
in the context of COMPAS!

# Hands-On Data Science!! :0



## Next Steps:

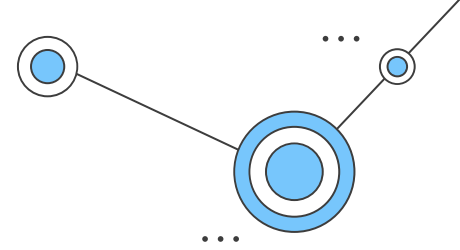
1. Find the F25 CRA repo in the [MDST GitHub](#) and download all the relevant files in the Week 6 directory (kaplan\_meier\_blank.ipynb, compas-KM-data.csv)
2. Split into teams of 2-3 (new ones or same as last week)
  - a. If you're working with new people, introduce yourselves!!
  - b. Start talking about your final presentation topic!!
3. Work on the exercises in the notebooks!
  - a. Ask us if you need help!



[Pandas Cheat Sheet](#)

[Seaborn Cheat Sheet](#)

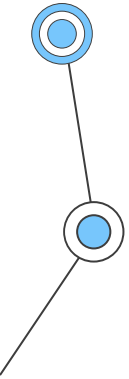
# Reminders



- Don't share colab notebooks with teammates if you are working at the same time
- Where to put csv and data files
  - Google Drive
    - Need to include:

```
from google.colab import drive
drive.mount('/content/drive')

pd.read_csv('/content/drive/MyDrive/[FILE NAME]')
```
  - Colab Files
    - See next slides





# Reminders

Click on the  
folder in the  
sidebar



The screenshot shows a JupyterLab environment with a sidebar on the left and a main workspace. The sidebar has a 'Files' tab with a folder icon and a list containing '..' and 'sample\_data'. A red arrow points to the 'sample\_data' folder. The main workspace has a title bar with 'Commands', '+ Code', '+ Text', and 'Run all'. Below the title bar is a toolbar with icons for file operations. The main area is titled 'Week 1 - Pandas Practice' and contains the following content:

Here is where you import the libraries necessary to perform the following tasks!

```
[ ] import pandas as pd
import seaborn as sns

# Allows you to provide a path to a Google Drive address rather than a local file path
from google.colab import drive
drive.mount('/content/drive')
```

Mounted at /content/drive

Load the Google Forms .csv into a Pandas dataframe.

```
[ ] df = pd.read_csv('/content/MDST Week 1 - Pandas Practice.csv')
```

Print out the .head() and the datatypes.

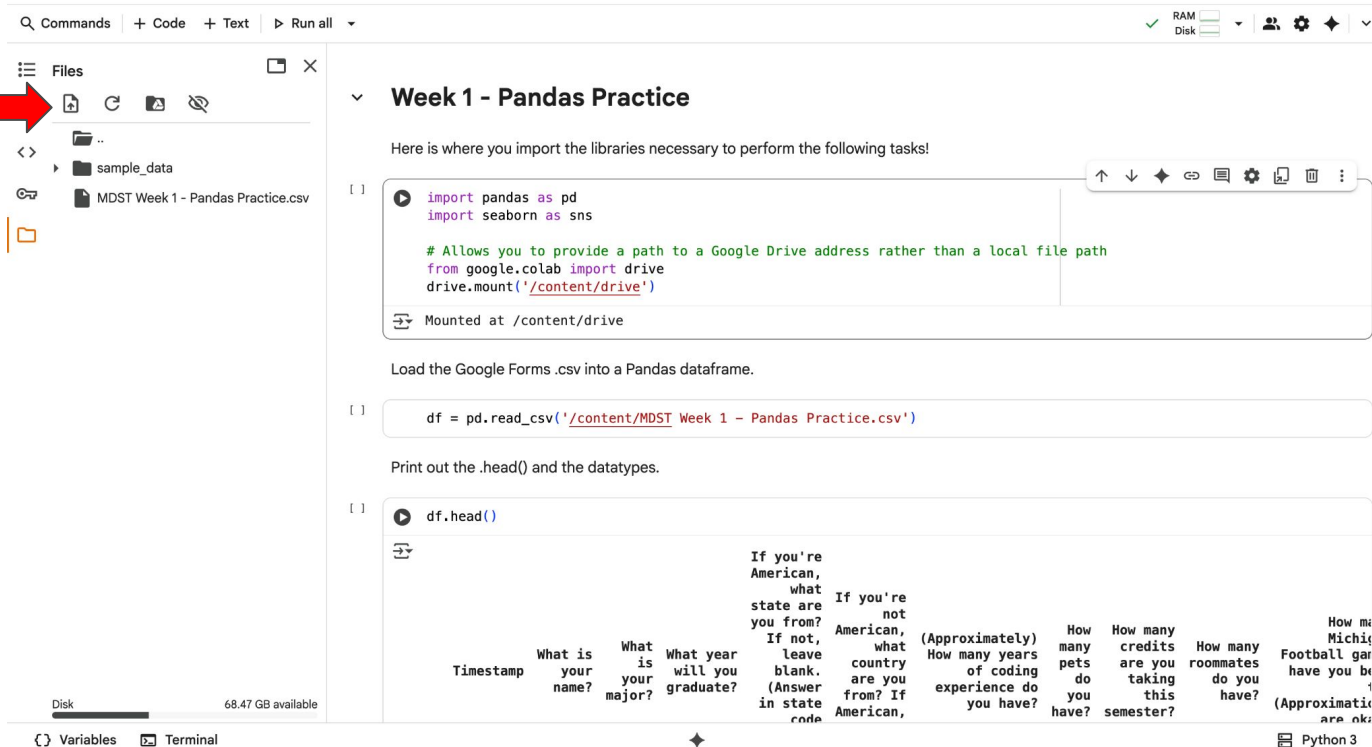

```
[ ] df.head()
```

The output of the code is a table with the following columns: Timestamp, What is your name?, What is your major?, What year will you graduate?, If not, leave blank. (Answer in state code), If you're American, what state are you from?, If you're not American, what country are you from?, (Approximately) How many years of coding experience do you have?, How many pets do you have?, How many credits are you taking this semester?, How many roommates do you have?, How many Football games have you been to?, and How many Michigan games have you been to? (Approximately how many are okay?).

At the bottom of the interface, there is a status bar showing 'Disk' with a progress bar and '68.47 GB available', and a 'Variables' tab. The bottom right corner shows 'Python 3'.

# Reminders

Click the upload button and select the file you want to upload



The screenshot shows the Google Colab interface. On the left, the 'Files' pane shows a folder named 'sample\_data' and a file named 'MDST Week 1 - Pandas Practice.csv'. A red arrow points to the upload button (a square with a plus sign) in the file explorer. The main area displays a Jupyter Notebook titled 'Week 1 - Pandas Practice'. The notebook contains the following code:

```
import pandas as pd
import seaborn as sns

# Allows you to provide a path to a Google Drive address rather than a local file path
from google.colab import drive
drive.mount('/content/drive')
```

Below the code, it says 'Mounted at /content/drive'. The next cell contains the following code:

```
df = pd.read_csv('/content/MDST Week 1 - Pandas Practice.csv')
```

Below the code, it says 'Load the Google Forms .csv into a Pandas dataframe.' The next cell contains the following code:

```
df.head()
```

Below the code, it shows the output of the `df.head()` command, which is a preview of the data from the CSV file. The output is a table with columns: Timestamp, What is your name?, What is your major?, What year will you graduate?, If you're American, what state are you from?, If you're not American, what country are you from?, (Approximately) How many years of coding experience do you have?, How many pets do you have?, How many credits are you taking this semester?, How many roommates do you have?, and How many Michigan Football games have you been to? (Approximately).

# Reminders

Click the three dots and copy the path. Put this in your read function



The screenshot shows a Google Colab interface. On the left, a file explorer shows a folder named 'sample\_data' with a subfolder 'MDST'. A context menu is open over the 'MDST' folder, with 'Copy path' selected. A red arrow points from the text 'Click the three dots and copy the path. Put this in your read function' to the 'Copy path' option. The main code editor shows the following code:

```
[ ] import pandas as pd
import seaborn as sns

# Allows you to provide a path to a Google Drive address rather than a local file path
from google.colab import drive
drive.mount('/content/drive')

Mounted at /content/drive

Load the Google Forms .csv into a Pandas dataframe.

[ ] df = pd.read_csv('/content/MDST Week 1 - Pandas Practice.csv')

Print out the .head() and the datatypes.

[ ] df.head()
```

The output of the code shows the following data:

Timestamp	What is your name?	What is your major?	What year will you graduate?	If not, leave blank. (Answer in state code)	If you're American, what state are you from?	If you're not American, what country are you from? If American, (Approximately) How many years of coding experience do you have?	How many pets do you have?	How many credits are you taking this semester?	How many roommates do you have?	How many Football games have you been to? (Approximately are ok)

The interface also shows a 'Variables' panel at the bottom left and a 'Terminal' panel at the bottom right. The RAM and Disk usage is shown at the top right.