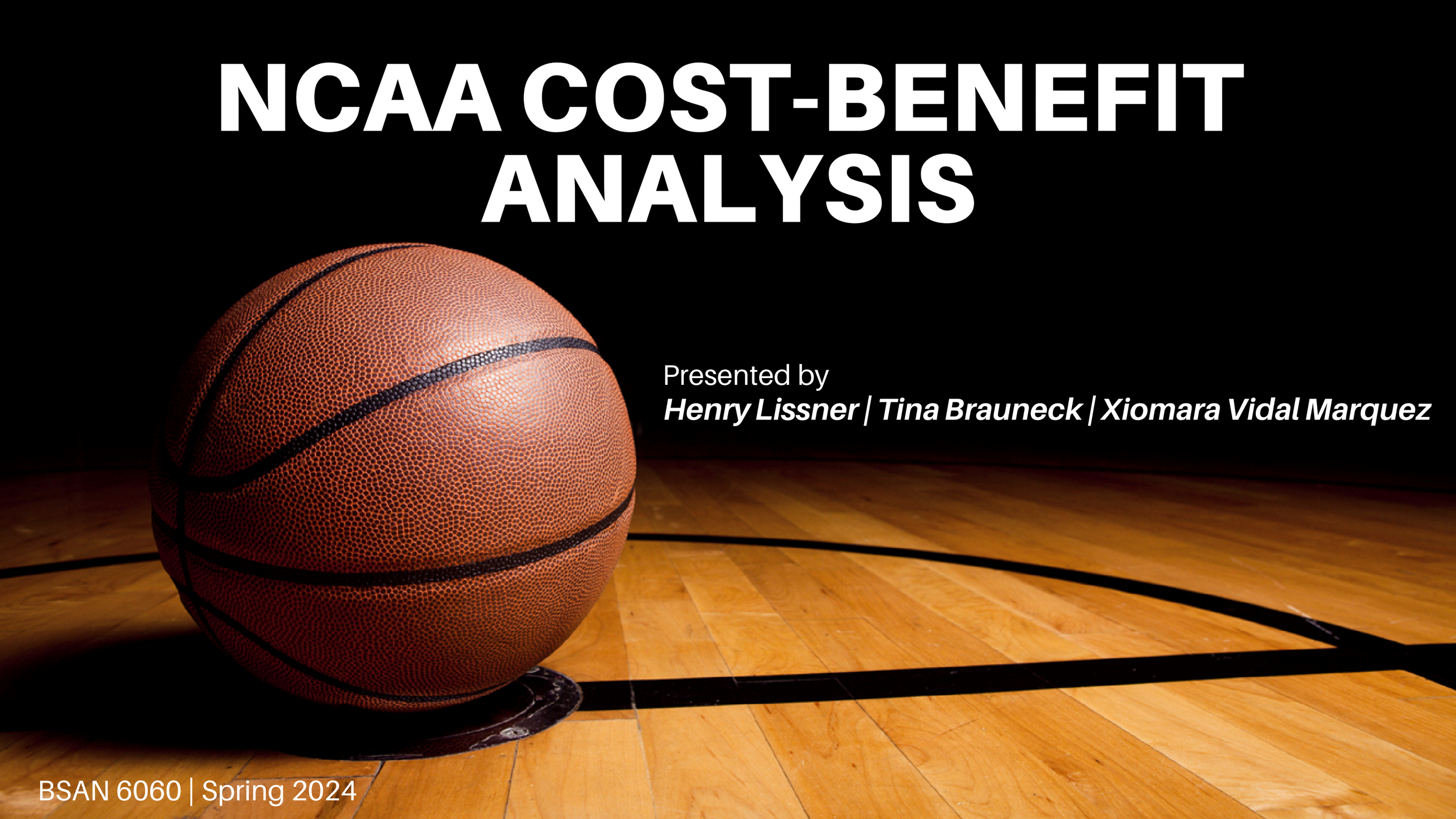
NCAA COST-BENEFIT ANALYSIS



Presented by
Henry Lissner | Tina Brauneck | Xiomara Vidal Marquez

BSAN 6060 | Spring 2024

BREAKING: "Save LMU Sports" protests LMU Athletics' decision to discontinue six sports

LOCAL NEWS

Thousands sign petition after LMU cuts 6 sports programs

Rodriguez

11:14 PM PST / KCAL News

End of Era for US Track and Field Team as LMU Athletics Cancels Six Programs Post NCAA Season

Published 01/25/2024, 3:15 AM PST

By SHAYNI MAITRA

BREAKING: LMU Athletics to discontinue six sports after 2023-24 season

Amy Carlyle, editor-in-chief Jan 23, 2024 Updated Jan 23, 2024 0

Loyola Marymount Drops Women's Swimming as Six Program Cut

Introduction

What we lost

This year, LMU discontinued six of its sports.

"This decision best positions our remaining WCC sports for greater success."

-Athletic Director Craig Pintens

Introduction

What we lost

For Division I status, NCAA requires schools:

- Sponsor at least 14 varsity sports, with at least 7 men's and 7 women's teams
- Provide athletic scholarships proportional to the participation rates of male and female students.

By discontinuing certain programs, LMU can focus on sports that have the potential for greater success.

But what if we could manage our resources better?

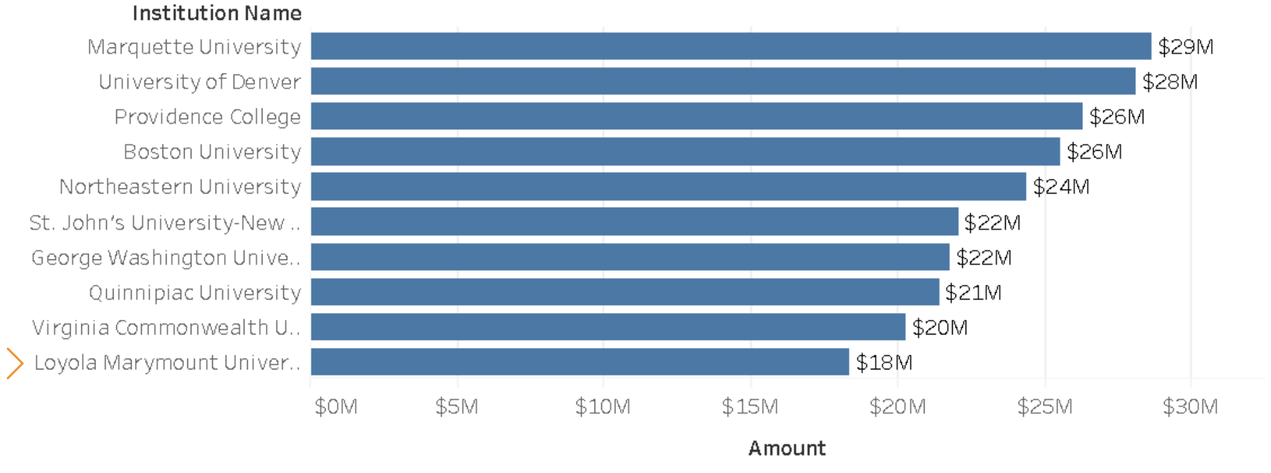
Goal

Does more $ = more wins?

Our project uses descriptive analytics to help answer this question. Understanding how expeditures, revenue, and performance are related can help inform and optimize resource management.

Report Intro in Notes

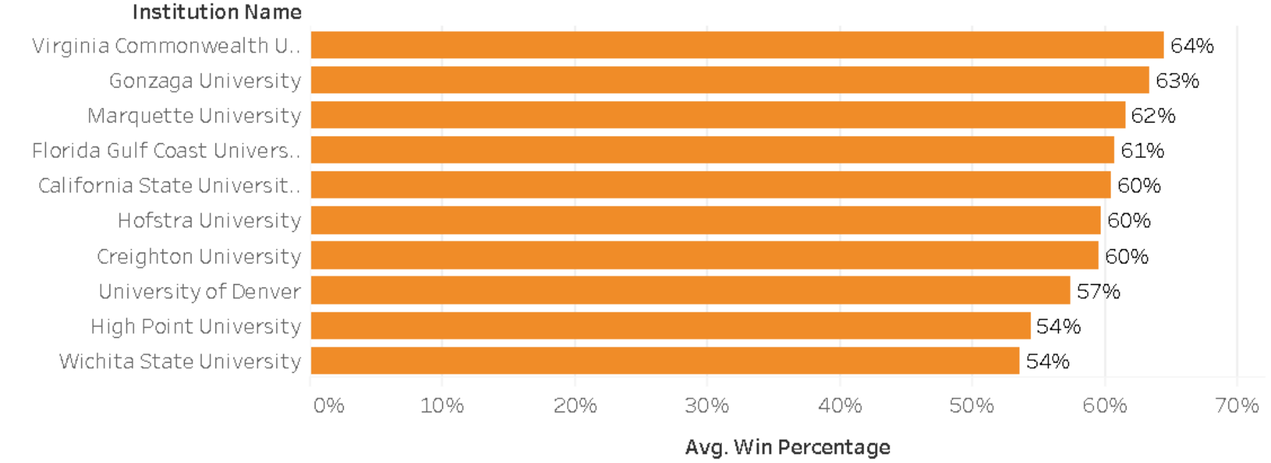
Top Spenders - 2021



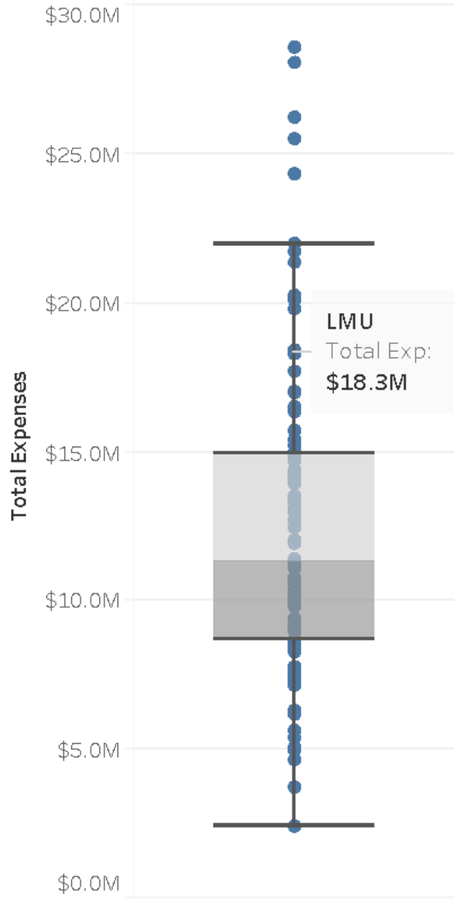
Expense Type

- [ ] assist_coach_salary_by_FTE
- [ ] head_coach_salary_by_FTE
- [ ] operating_expense_per_team
- [ ] recruitment
- [ ] student_aid
- [x] total_expense

Top Performing Institutions by Win Ratio - 2021



Sports Expense Distribution - 2021



In 2021, LMU ranked 10th in its division for overall athletic expenditures, spending well above the median. However, their win ratio for top sports is only 45%, well below the cut-off for top 10 performers.

In fact, many names on the top spender list do not appear on the top performer list, suggesting no direct link between more spending and increased athletic performance.

Scouting

Data Collection

Two Sources:

1. Schools' Sports Financial Data:
    - Equity in Athletics Disclosure Act Survey
    - https://ope.ed.gov/athletics/#/

2. NCAA Win-Loss Data:
    - NCAA Statistics
    - https://stats.ncaa.org/rankings/ranking_summary

Making the cut

Slicing and Loading

Scope Selection: NCAA Division I without football
(LMU's division)

Steps in ETL:

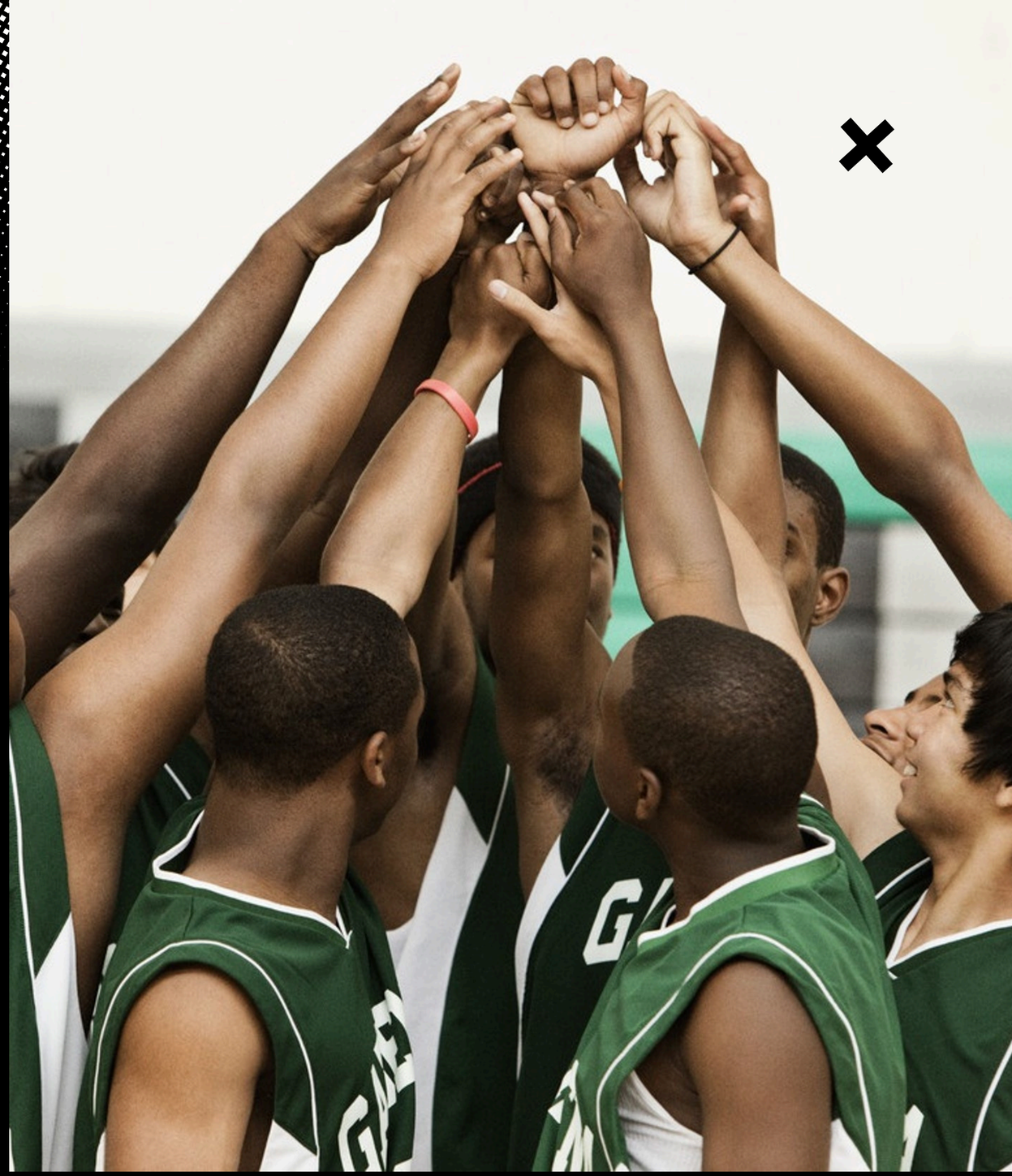
- Reading into python
- Union: all years
- Identified/Selected Features
- Sliced into separate dataframes
- Added columns with identifiers
- Read to SQL

Bringing it in

Entity Relationships

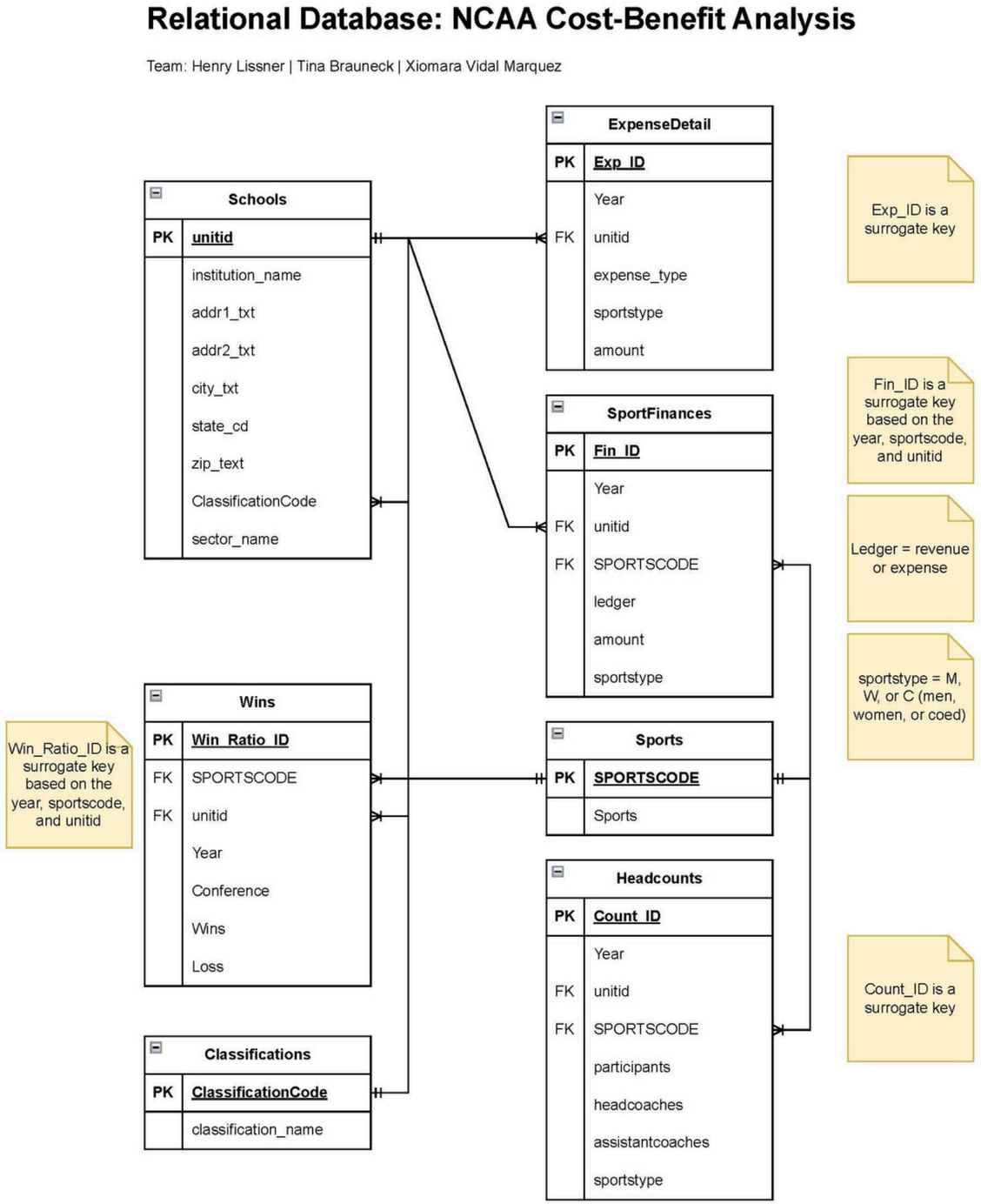
Step 1: Relational Database

Step 2: Data Warehouse



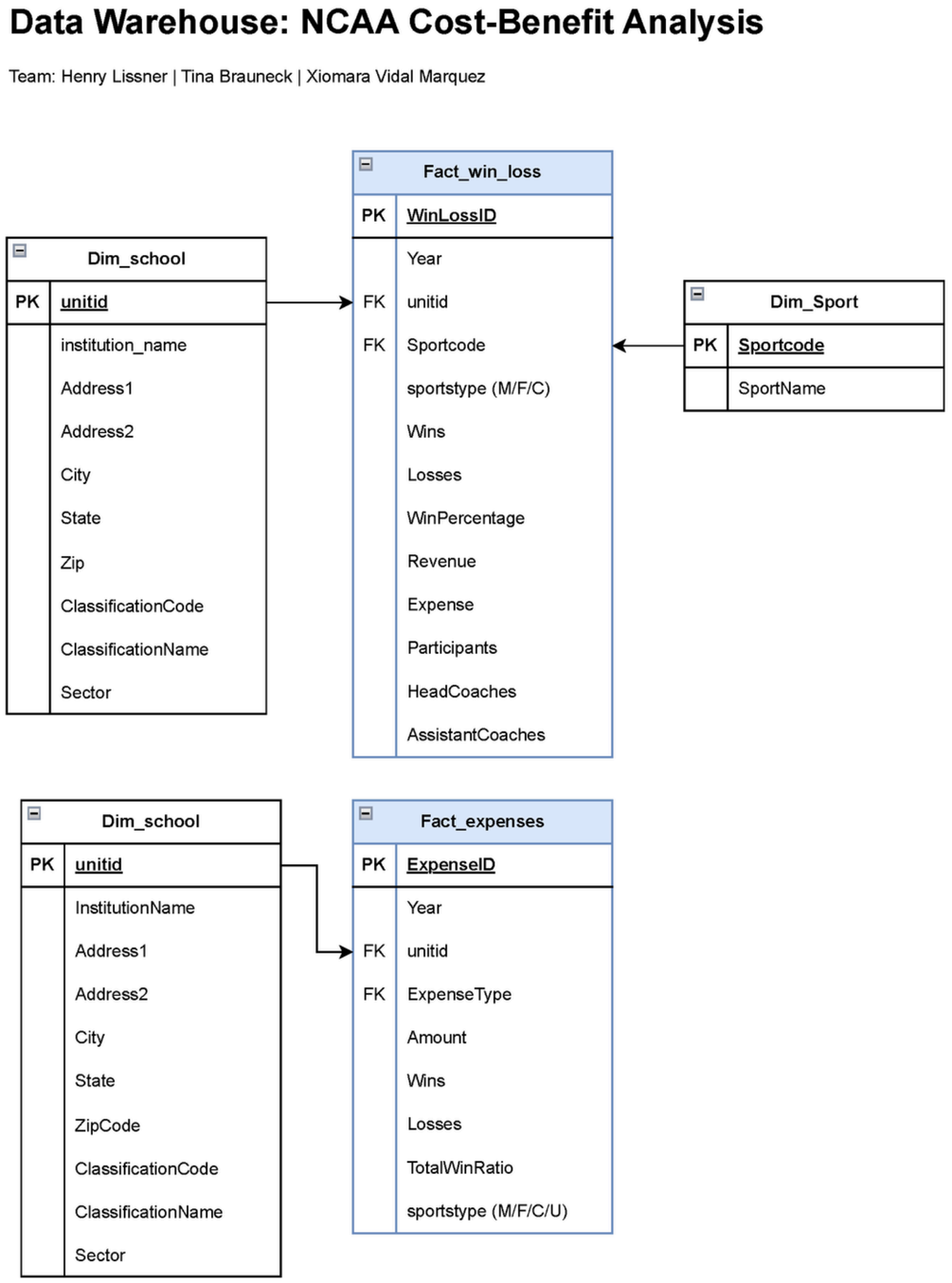
The Game Plan

Entity Relationship Diagrams (ERD)



The Game Plan

Entity Relationship Diagrams (ERD)



Data Gymnastics

Python & ETL

BSAN 6060 Project: NCAA Cost - Benefit Analysis

Team: Henry Lissner | Xiomara Vidal Marquez | Tina Brauneck

Spring 2024

In this file we load the data from each year, combine it, and then separate the fields into logical entities.

Import/Install

In [1]:
```
pip install mysql-connector-python
```

```
Requirement already satisfied: mysql-connector-python in c:\users\ttesn\anaconda3\lib\site-packages (8.3.0)
Note: you may need to restart the kernel to use updated packages.
```

In [2]:
```
pip install sqlalchemy
```

```
Requirement already satisfied: sqlalchemy in c:\users\ttesn\anaconda3\lib\site-packages (1.4.39)Note: you may need to restar
he kernel to use updated packages.

Requirement already satisfied: greenlet!=0.4.17 in c:\users\ttesn\anaconda3\lib\site-packages (from sqlalchemy) (2.0.1)
```

In [240]:
```
# get the required modules

from sqlalchemy import create_engine    # sqlalchemy connector
import os                               # get the environment variables
import pandas as pd                     # get the pandas
import numpy as np                      # get numpy
```

Connection Setup

```python
# List of file names
file_names = ['Womens Basketball 21-22 NCAA Statistics.xlsx', 'Womens Basketball 20-21 NCAA Statistics.xlsx',
              'Womens Basketball 19-20 NCAA Statistics.xlsx', 'Womens Basketball 18-19 NCAA Statistics.xlsx',
              'Womens Basketball 17-18 NCAA Statistics.xlsx', 'Womens Basketball 16-17 NCAA Statistics.xlsx',
              'Womens Basketball 15-16 NCAA Statistics.xlsx', 'Womens Basketball 14-15 NCAA Statistics.xlsx',
              'Womens Basketball 13-14 NCAA Statistics.xlsx', 'Womens Basketball 12-13 NCAA Statistics.xlsx',
              'Womens Basketball 11-12 NCAA Statistics.xlsx', 'Womens Basketball 10-11 NCAA Statistics.xlsx']

# Initialize an empty list to store DataFrames
dfs = []

# Iterate through each file and read into a DataFrame
for file_name in file_names:
    try:
        # Extract year from filename and prepend "20" to it
        year = "20" + file_name.split(' ')[-3].split('-')[0]

        # Read Excel file into DataFrame
        df = pd.read_excel(file_name)

        # Add a column for the year extracted from the filename
        df['Year'] = year

        # Append the DataFrame to the dfs list
        dfs.append(df)
    except Exception as e:
        print(f"Error reading {file_name}: {e}")

# Concatenate all DataFrames in the list
df = pd.concat(dfs, ignore_index=True)

# Write the merged DataFrame to a new Excel file
df.to_excel("merged_data_WBB.xlsx", index=False)

print("Merged data saved to merged_data.xlsx")
```

- Once each team's win loss data was merged for all 10 years we had the bases for the Win table

- Taking the datasets and merging them into one with the addition of a year column

```python
# Load the mapping from the CSV file
mapping_df = pd.read_csv('/Users/xiomara/Desktop/BSAN 6060/Project/Data/Agg/Merged/sportscode_xlsx.csv', header=None)
mapping_dict = pd.Series(mapping_df[1].values,index=mapping_df[0]).to_dict()


file_names = ['merged_data_BaseB.xlsx', 'merged_data_MBB.xlsx',
              'merged_data_MSo.xlsx', 'merged_data_MVB.xlsx',
              'merged_data_SoftB.xlsx', 'merged_data_WBB.xlsx',
              'merged_data_WSo.xlsx', 'merged_data_WVB.xlsx']

dfs = []

for file_name in file_names:
    try:
        df = pd.read_excel(file_name)

        # Rename columns if needed
        df.rename(columns={'ï»¿Rank':'Rank','W': 'Won', 'L': 'Loss','Lost':'Loss' ,'T': 'Tied'}, inplace=True)

        base_file_name = file_name.split('/')[-1]

        if base_file_name in mapping_dict:
            df['Code'] = mapping_dict[base_file_name]
        else:
            print(f"No code found for {base_file_name}. Skipping.")

        # Append DataFrame to list
        dfs.append(df)
    except Exception as e:
        print(f"Error reading {file_name}: {e}")

# Concatenate all DataFrames in the list
df = pd.concat(dfs, ignore_index=True)

# Save the merged data to an Excel file
df.to_excel("merged_data.xlsx", index=False)

print("Merged data saved to 'merged_data.xlsx'.")
```

```
# declare starting year as variable

year = 2010
```

```
# initialize an empty list for years

years = []
```

```
# Create a list of years included using a loop

for x in range(12):
    loop_year = year + x
    years.append(loop_year)
```

```
print(years)
```

```
[2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021]
```

```
# initialize a dataframe for all years, starting with the cleaned version of 2022

all_years_df = Cleaned_AY21_22_df
```

```
# append the all_years_df with each year of data

for y in years:
    file_name_str = str(y)+'_schools.sas7bdat'
    temp_df = pd.read_sas(file_name_str, encoding = 'latin-1')
    temp_df.insert(0, "year", y)
    all_years_df = pd.concat([all_years_df, temp_df])
```

```
headcount_df = Schools_all_years_df.iloc[:,np.r_[0,1,16,17,18]]
```

```
headcount_df = headcount_df.assign(COED = headcount_coedpart_df)
```

```
headcount_df.fillna(0)
```

	year	unitid	SPORTSCODE	PARTIC_MEN	PARTIC_WOMEN	COED
0	2022	100654.0	1.0	37.0	0.0	0.0
1	2022	100654.0	2.0	14.0	17.0	0.0
11	2022	100654.0	33.0	0.0	8.0	0.0
9	2022	100654.0	25.0	0.0	6.0	0.0
2	2022	100654.0	7.0	96.0	0.0	0.0
...	...	...	...	...	...	...
16478	2010	446048.0	15.0	16.0	20.0	0.0
16479	2010	446048.0	16.0	0.0	13.0	0.0
16480	2010	446048.0	26.0	0.0	9.0	0.0
16481	2010	446233.0	1.0	12.0	0.0	0.0
16482	2010	446233.0	2.0	13.0	12.0	0.0

```
sportstype_list = ['M','W','C']
```

```
cn = 'Participants' #change expense type here
df_at_start_loop = headcount_df.copy() #change data dataframe here
st = 0 # sprorts type list index

#This loop goes for an extra round because we have an unallocated sports type
for c in range(3,6):
    df = df_at_start_loop.iloc[:,np.r_[0,1,2,c]]
    df = df.assign(sportstype = sportstype_list[st])
    st = st+1
    df = df.rename(columns={df.columns[3]:cn})
    headcounts_all_df = pd.concat([headcounts_all_df,df])
```

```
headcount_headcoach_df = Schools_all_years_df.iloc[:,np.r_[0,1,16,46,55,64]]
```

```
headcount_headcoach_df.head()
```

	year	unitid	SPORTSCODE	MEN_TOTAL_HEADCOACH	WOMEN_TOTAL_HDCOACH	COED_TOTAL_HDCOACH
0	2022	100654.0	1.0	1.0	NaN	NaN
1	2022	100654.0	2.0	1.0	1.0	NaN
11	2022	100654.0	33.0	NaN	1.0	NaN
9	2022	100654.0	25.0	NaN	1.0	NaN
2	2022	100654.0	7.0	1.0	NaN	NaN

```
headcoach_df = pd.DataFrame()
```

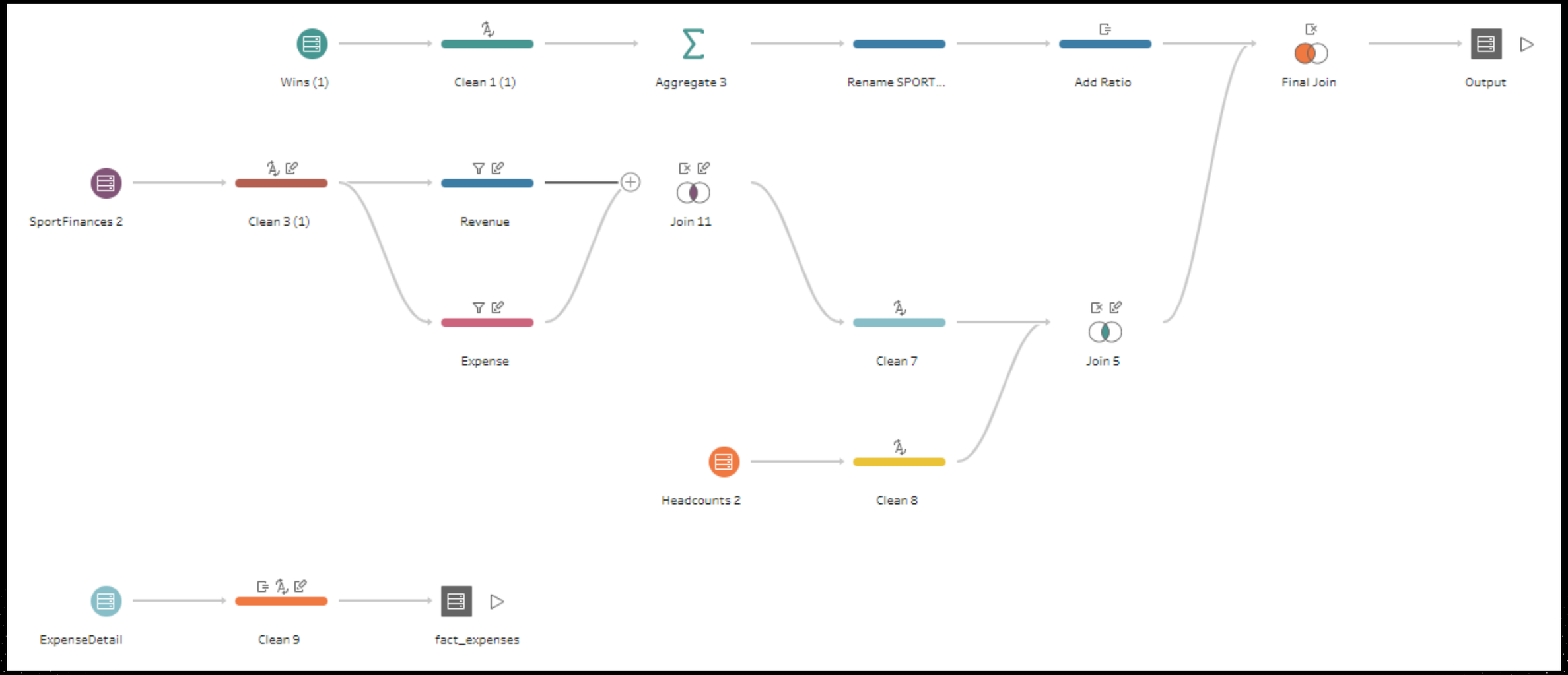
```
cn = 'HeadCoaches' #change expense type here
df_at_start_loop = headcount_headcoach_df.copy() #change data dataframe here
st = 0 # sprorts type list index

#This loop goes for an extra round because we have an unallocated sports type
for c in range(3,6):
    df = df_at_start_loop.iloc[:,np.r_[0,1,2,c]]
    df = df.assign(sportstype = sportstype_list[st])
    st = st+1
    df = df.rename(columns={df.columns[3]:cn})
    headcoach_df = pd.concat([headcoach_df,df])
```

```
headcoach_df
```

	year	unitid	SPORTSCODE	HeadCoaches	sportstype
0	2022	100654.0	1.0	1.0	M
1	2022	100654.0	2.0	1.0	M
11	2022	100654.0	33.0	NaN	M
9	2022	100654.0	25.0	NaN	M
2	2022	100654.0	7.0	1.0	M
...	...	...	...	...	...
16478	2010	446048.0	15.0	NaN	C
16479	2010	446048.0	16.0	NaN	C
16480	2010	446048.0	26.0	NaN	C
16481	2010	446233.0	1.0	NaN	C
16482	2010	446233.0	2.0	NaN	C

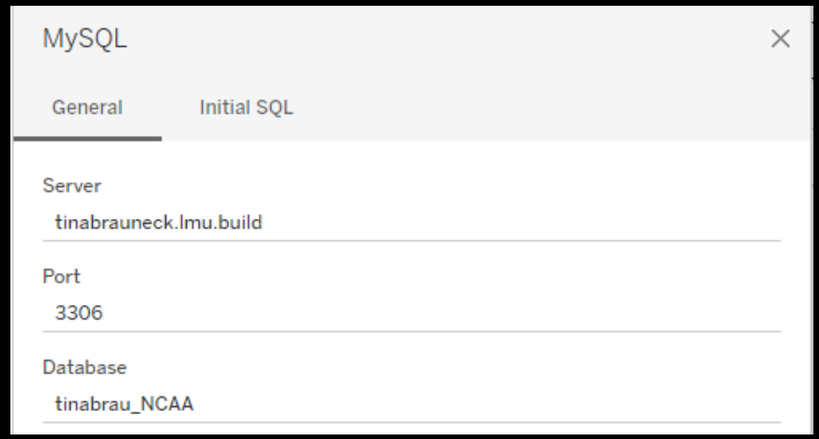
796746 rows × 5 columns

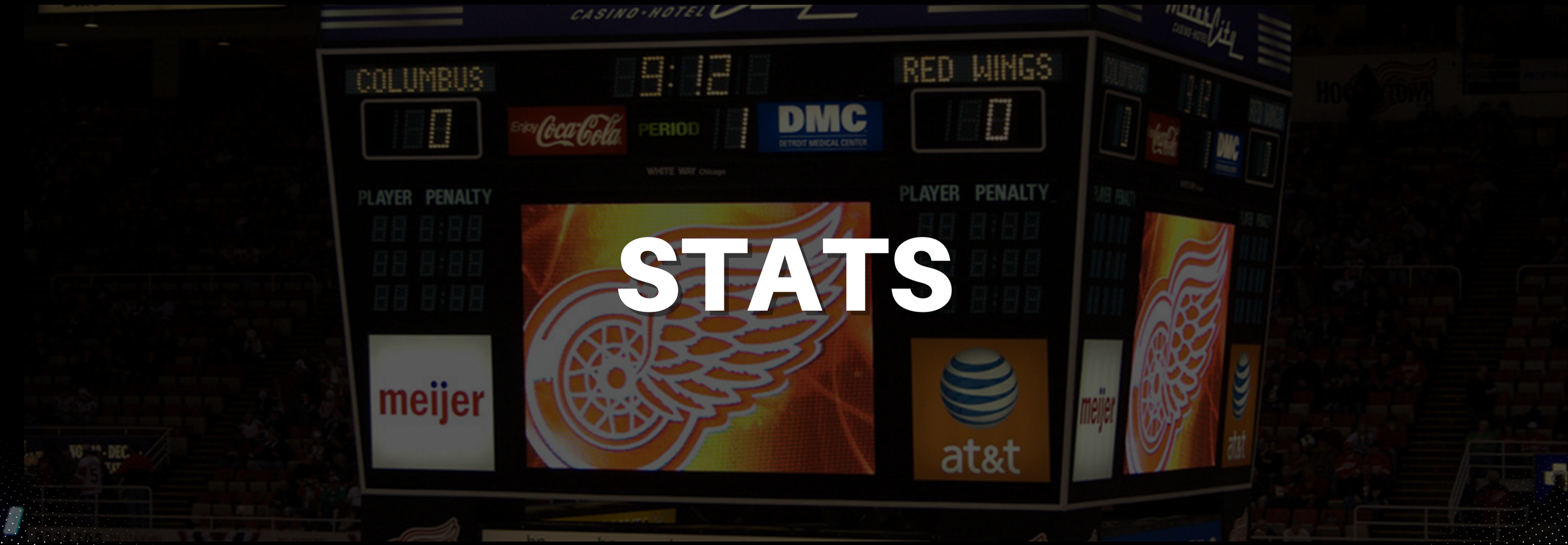

Wins (1)
Clean 1 (1)
Aggregate 3
Rename SPORT...
Add Ratio
Final Join
Output
SportFinances 2
Clean 3 (1)
Revenue
Join 11
Expense
Clean 7
Join 5
Headcounts 2
Clean 8
ExpenseDetail
Clean 9
fact_expenses

Time to Play

Visualizing in Tableau

We connected directly to our data warehouse in Tableau. After this step, it was game on!




CASINO·HOTEL
COLUMBUS
0
9:12
PERIOD
1
DMC
DETROIT MEDICAL CENTER
RED WINGS
0
PLAYER PENALTY
PLAYER PENALTY
meijer
at&t
STATS

STATS

Institution's Overall Sports Expenses

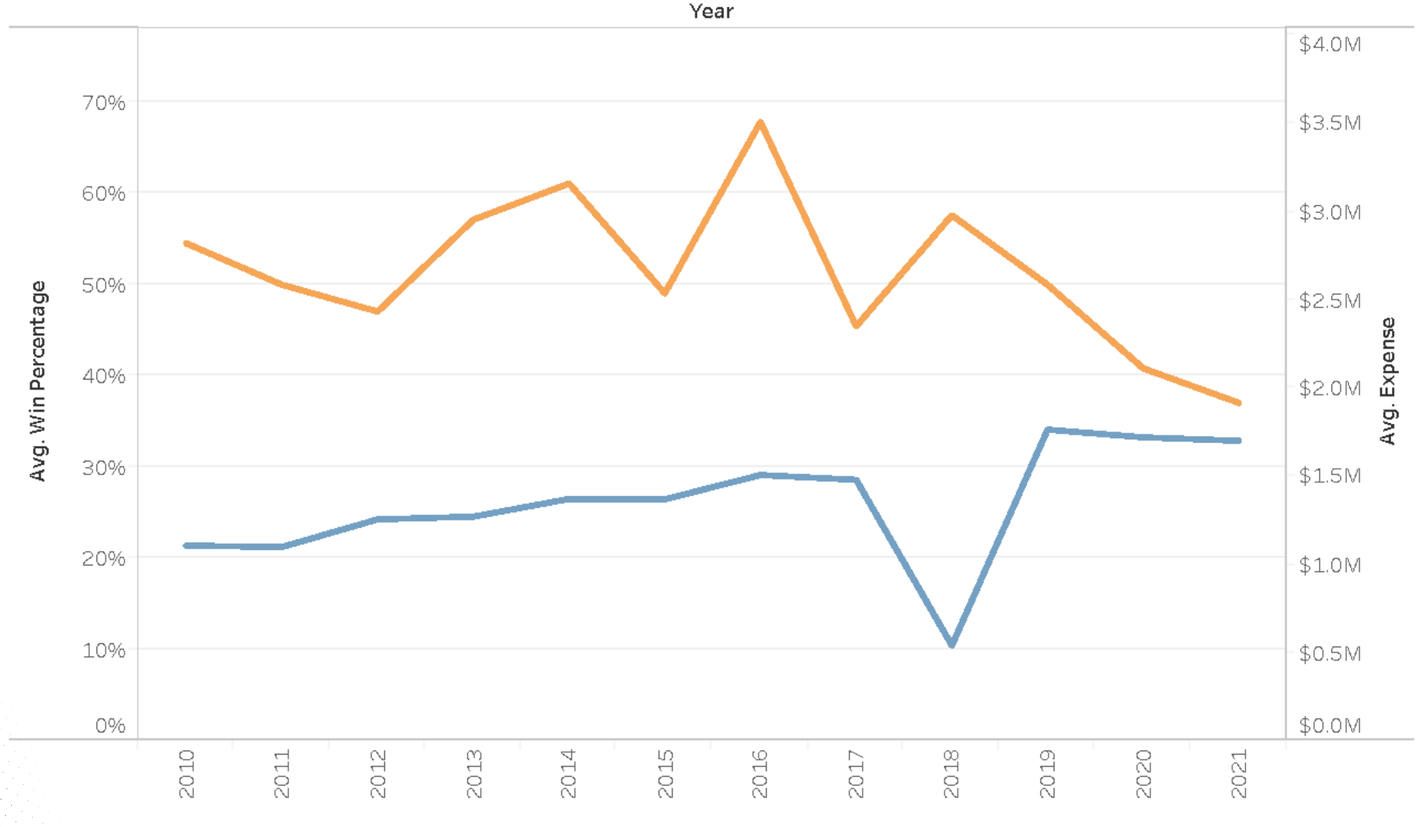
Expense Type	2010	2011	2012	2013	2014	2015	2016	2017
	Year							
assist_coach_salary_by_FTE	$115K	$121K	$116K	$120K	$124K	$124K	$118K	$11
head_coach_salary_by_FTE	$175K	$185K	$189K	$195K	$208K	$230K	$230K	$23
operating_expense_per_team	$1,709K	$1,894K	$2,219K	$2,114K	$2,172K	$2,387K	$2,567K	$2,66
recruitment	$199K	$229K	$235K	$276K	$319K	$378K	$385K	$38
student_aid	$7,440K	$7,414K	$7,857K	$7,958K	$8,166K	$8,499K	$8,861K	$8,82
total_expense	$14,567K	$14,511K	$15,974K	$15,926K	$16,586K	$17,217K	$17,632K	$18,05

Institution Name
Loyola Marymount Unive..

Take a deeper dive into an institution's expenses. Select an intitution name for each chart. The table at the top shows the different types of expenditures for the institution. The bottom graph compares the win-loss ratio year-over-year to the expenses, sport-by-sport.

We can see in the bottom chart that there is a lot of random variation in the win-loss ratios and the expenses don't trend similarly in most cases.

Win Ratio vs Expenses Year-Over-Year - Baseball, Loyola Marymount University



Institution Name
Loyola Marymount Unive..

Limit
Top 10 by AVG([Win Percentage])

Sportname
Baseball
☐ Show history

Measure Names, Instituti..
Avg. Expense, Loyola ..
Avg. Win Percentage,..

Map - Basketball

Avg. Win Percentage

5%
20%
40%
60%
84%

Avg. Win Percentage

5% 84%

Sportname

Basketball

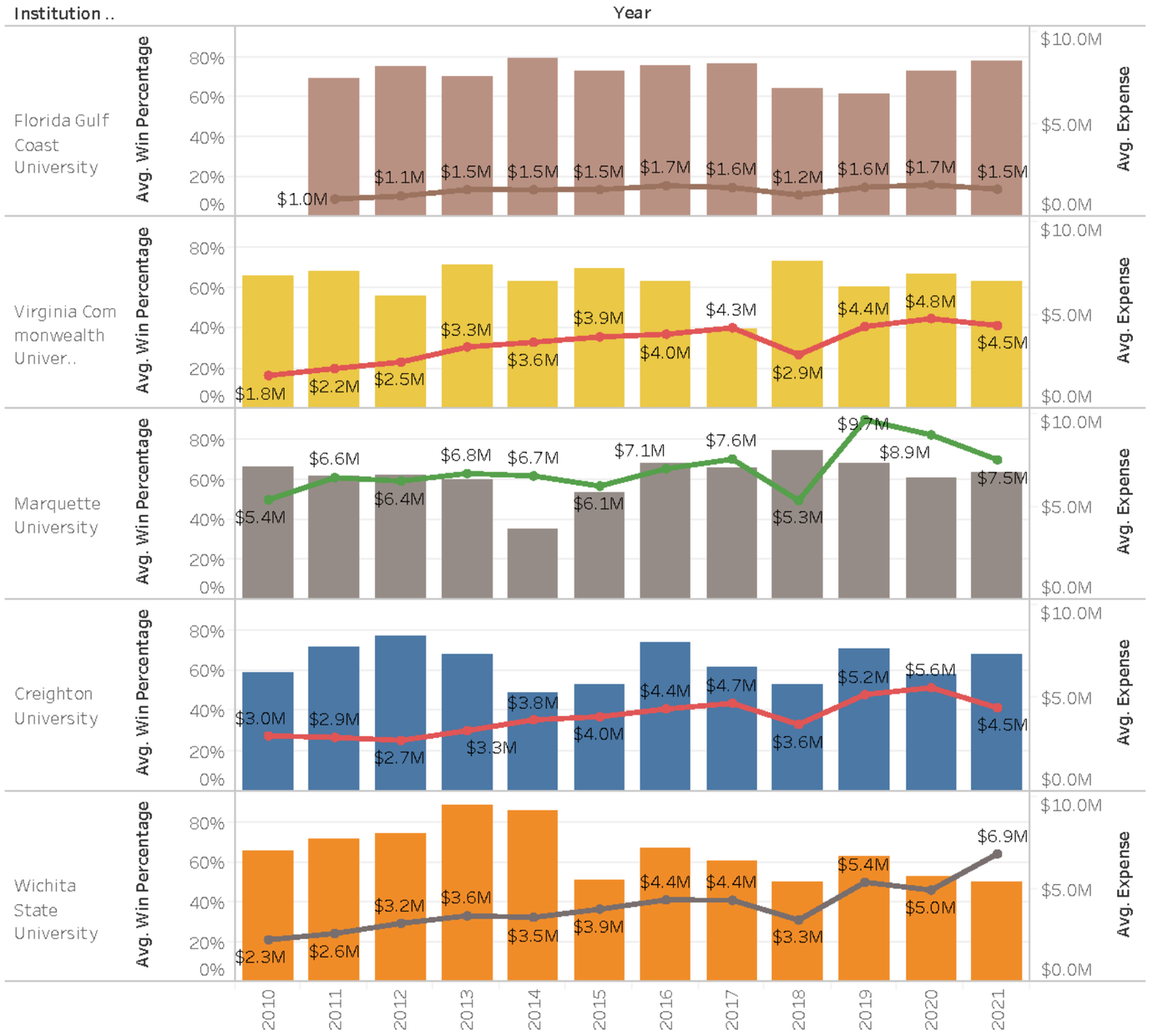
Show history

There was no notable trend in win ratios by location.

This suggests LMU gains no advantage from its warmer, drier climate.

User tip: use the page toggle to see a map for each sport.

Top 5 Instituitons - Basketball



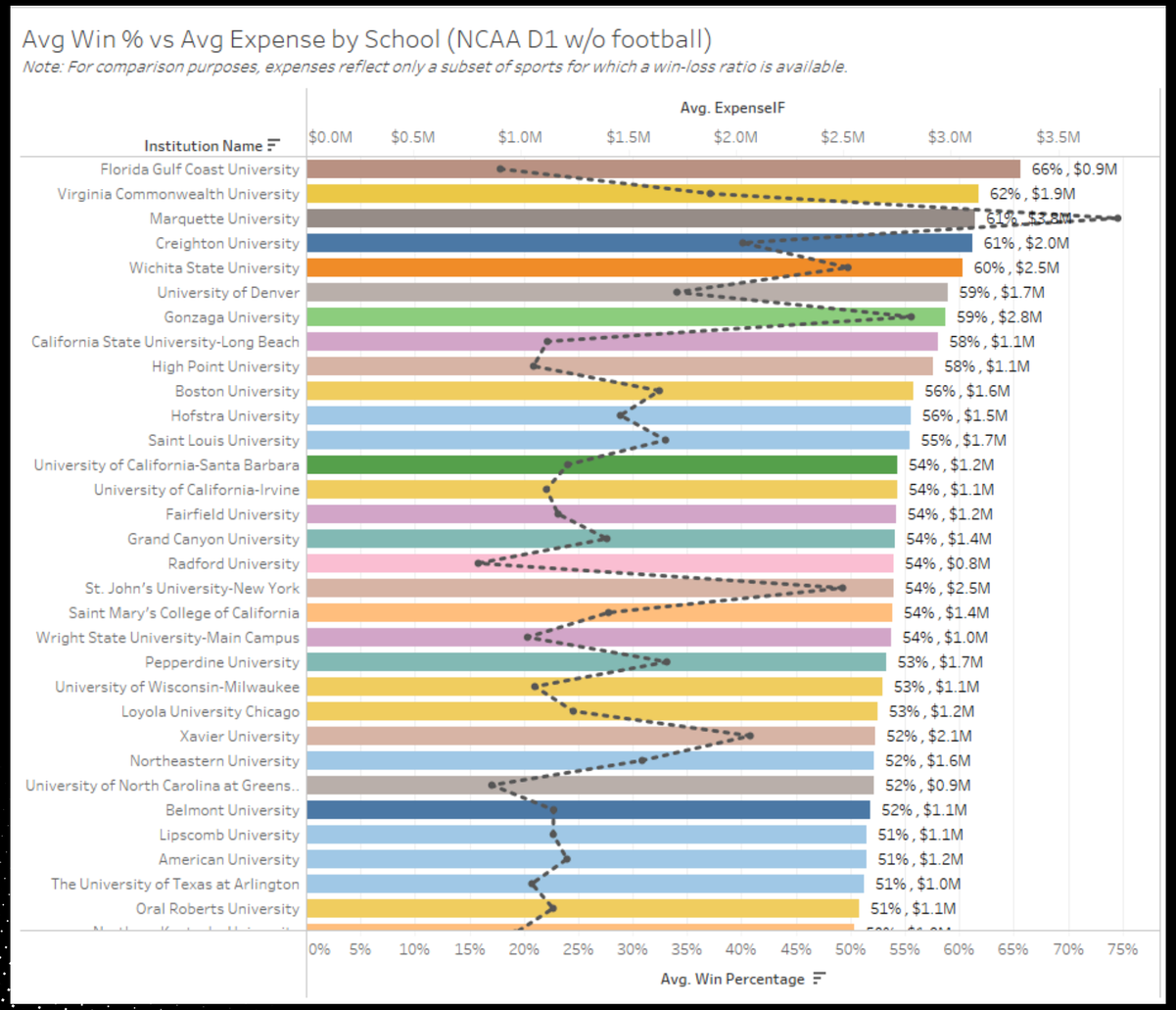
Sportname
Basketball
Show history

To the left are the top 5 instituitions in terms of win ratio. We can see from this graph that spending trends do not align with performance. There is a lot of random variation.

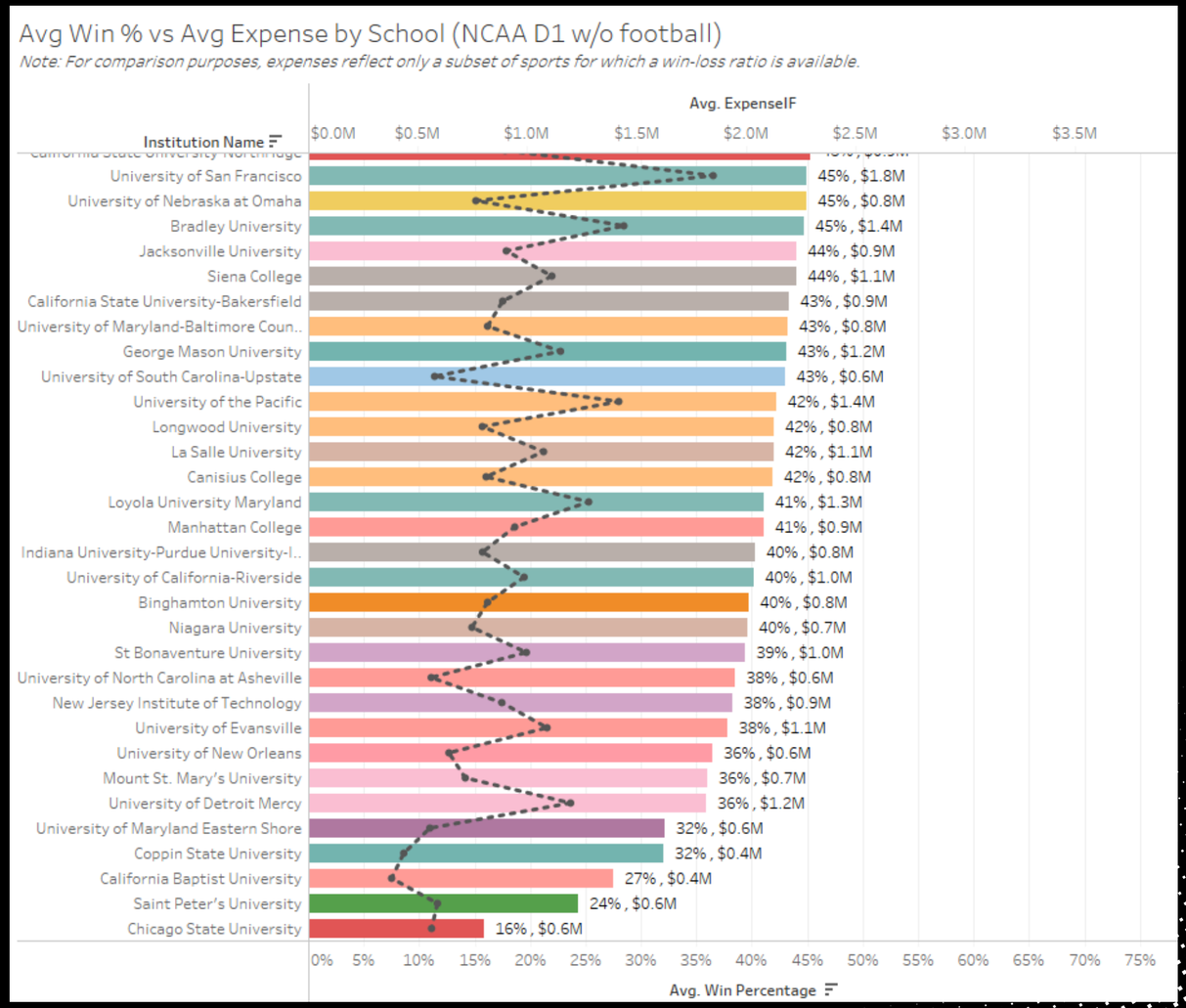
User tip: Page through different sports using the Sportname box above.

Average Win Percentage and Average Expenses

Top Institutions



Bottom Institutions



Next Steps

Future Exploration

Now that the warehouse has been generated and we've visualized our initial findings, this is a good stopping point to check in with stakeholders and evaluate next steps.

- Possible investigations:
  - How do head coach and assistant coach salaries vary by location?
  - What do top performers spend on recruitment? Low performers?
  - How do expenses for women's sports compare to men's?
  - Can machine learning predict win ratios based on the data?

Working through the pain

Challenges

- Fragmented data
  - Our data not only came from multiple sources, but was split into multiple files -- one per year or in some cases, one per sport per year
  - The data needed to be mapped with a common identifier (i.e., school name), since the source data did not match
- Incomplete data
  - Win-loss records could not be located for some sports.
- Getting the right schema design
  - Choosing how to design the fact tables can be challenging. Even during ETL we had to revise our design to avoid duplicating or losing data during joins.

Wins

Learnings

- ETL is time consuming! Plan accordingly.
- Python
  - pd.assign
  - pd.duplicates
  - pd.fillna
- Data base migration
  - MySQL workbench has a migration wizard that allows you to copy your database.

THANK YOU

Presentation by

Henry Lissner | Tina Brauneck | Xiomara Vidal Marquez