Chris McAllister

SIADS 521: Week 3 Assignment

Visualization Technique

In this 'article' we'll make 5 visualizations:

1. A simple line graph
2. Color-coded bar chart time series with an interactive tooltip
3. A box plot with a tooltip
4. Another box plot with an interactive tooltip.
5. An overlapping histogram.

The simple line graph is leveraged to showcase how our data is changing over time, and the second visual is an enhanced version of the line graph. The color-coded bar chart is essential because it allows us to add two more 'dimension' to the visual. The breakout by color allows us to see categorical data that the simple line graph doesn't show, and the tool tip adds an additonal variable to visual.

The box plots force us to sacrficie the time series component of the first two visuals, but instead we can understand how the same data is distributed across certain categories. The tool-tip allows the user to un-earth "row-level" insights from the dataset.

Lastly, the overlapping histogram offers a similar view to the boxplot, but with more customization (in terms of number of bins and what categorical items to keep in the plot).

I also considered leveraging a "Barley Trellis Plot" to show certain team rankings change between the start and end of the season, but I didn't think it would be necessary for the article.

Visual #2 is controversial. Typically, bar charts shouldn't be leveraged in time series analysis. However, I just didn't like how the plot looked with either an area chart or a line graph.

Visualization Library

For this article, we're going to leverage the Altair library.

Why Altair?

Our analysis requires us to explore data leveraging several different attributes at once: Year, Final AP Poll Rankings, Head Coach, and season win total. To include as many attributes as possible in a visual, I wanted to include a tool tip. In some examples below, we leverage both axes and color to show information, which means to only source method left is with an interactive tooltip. Altair is one of the few python data visualization libraries that allow you to use a tooltip.

Altair is EXTREMELY customizable. There aren't many visualization you can't create with it, and it comes with a massive gallery to quickly see what's possible: https://altair-viz.github.io/gallery/ (https://altair-viz.github.io/gallery/)

Why NOT Altair?

Altair can hit some speed bumps if your dataset is larger than 5,000 rows (note that our data set for this analysis is only ~150 rows). If you're working with large datasets, there are workarounds, see the following link from github: https://github.com/altair-viz/altair/issues/611 (https://github.com/altair-viz/altair/issues/611)

How to install Altair

Altair is open source, but you need to install it before you can just import it into a Jupyter Notebook.

If you're using anaconda, you can install it with following: conda install -c conda-forge altair

For those who prefer pip install: pip install altair vega_datasets

Altair Documentaion for more information: https://altair-viz.github.io/getting_started/installation.html (https://altair-viz.github.io/getting_started/installation.html)

Demonstration

For the demonstration, we'll be using a dataset from sports reference (https://www.sports-reference.com/cfb/schools/michigan/index.html (https://www.sports-reference.com/cfb/schools/michigan/index.html)) that summarizes Michigan Football seasons by a handful of attributes, such as: who coached the team, season win total, the team's final AP Poll Rankings, and the outcome of their final bowl game (if they played in one).

We are trying to determine whether Jim Harbaugh, the current head coach of our Michigan Wolverines, is a "good" coach by Michigan's very impressive historical standards.

In [1]:
```
# Import Libraries:
import pandas as pd
import numpy as np
import re

# Our data visualization library of choice: altair
import altair as alt
```

In [2]:
```
Data Source: 'https://www.sports-reference.com/cfb/schools/michigan/index.html'

Read in data (web page above saved as an html file)
ins = pd.read_html('Michigan_AP_Poll.html')
ins = pd.DataFrame(wins[0])

drop big10 season records. Create weird, duplicate columns names
ols = [7, 8, 9]
ins.drop(wins.columns[cols],axis=1,inplace=True)

Data cleaning, removing multi-layered axes
ins = wins.T
ins = wins.reset_index()
ins = pd.DataFrame(wins)
ins = wins.iloc[: , 1:]
ins = wins.T
ins.columns = wins.iloc[0]
ins = wins.iloc[1: , :]


Regex pattern to get JUST the name of coach, and create a new column called "Coach"
attern = '([A-Za-z]+\s{1}[A-Za-z]+[a-z]+)+.*'

Apply regex patter to the df
ins['Coach'] = wins['Coach(es)'].apply(lambda x: (re.findall(pattern, str(x))))
ins['Coach'] = wins['Coach'].str[0]

Drop columns we don't need
ols = [0, 7, 8, 9]
ins.drop(wins.columns[cols],axis=1,inplace=True)

Remove summary rows that pop up every 20 season
ins = wins[wins['W'] != 'W']

Create new column that sums the number of games played
ins['games_played'] = wins['W'].fillna('0').astype(int) + wins['L'].fillna('0').astype(int) + wins['T'].f

Create new column: wins_normalized

wins normalized is the number of wins the team would have had 13 game schedule (full 2021 schedule + a b
This is essential to compare against older teams that played fewer games
if team didn't qualify for a bowl game, then it counts it as a loss
Ties (which don't happen anymore) also count as losses now
ins['wins_normalized'] = (wins['W'].fillna('0').astype(int) / wins['games_played']) * 13

Create a similar df as "wins" but it only includes seasons after 1950 (called wins_1950)

This will upset some Michigan alumni, but if the team played with leather helmets
and used a literal pigskin for the ball then it's time to stop clinging to the past
ins_1950 = wins[wins['Year'] >= '1950']

Data preview
ins.head()
```

Out[2]:

level_1	Year	Conf	W	L	T	AP Pre	AP High	AP Post	Coach(es)	Bowl	Notes	Coach	games_played	wins_normalized
0	2021	Big Ten	8	1	NaN	NaN	6	NaN	Jim Harbaugh (8-1)	NaN	NaN	Jim Harbaugh	9	11.555556
1	2020	Big Ten	2	4	0	16	13	NaN	Jim Harbaugh (2-4)	NaN	NaN	Jim Harbaugh	6	4.333333
2	2019	Big Ten	9	4	0	7	7	18	Jim Harbaugh (9-4)	Citrus Bowl-L	NaN	Jim Harbaugh	13	9.000000
3	2018	Big Ten	10	3	0	14	4	14	Jim Harbaugh (10-3)	Peach Bowl-L	NaN	Jim Harbaugh	13	10.000000
4	2017	Big Ten	8	5	0	11	7	NaN	Jim Harbaugh (8-5)	Outback Bowl-L	NaN	Jim Harbaugh	13	8.000000

The Basics:

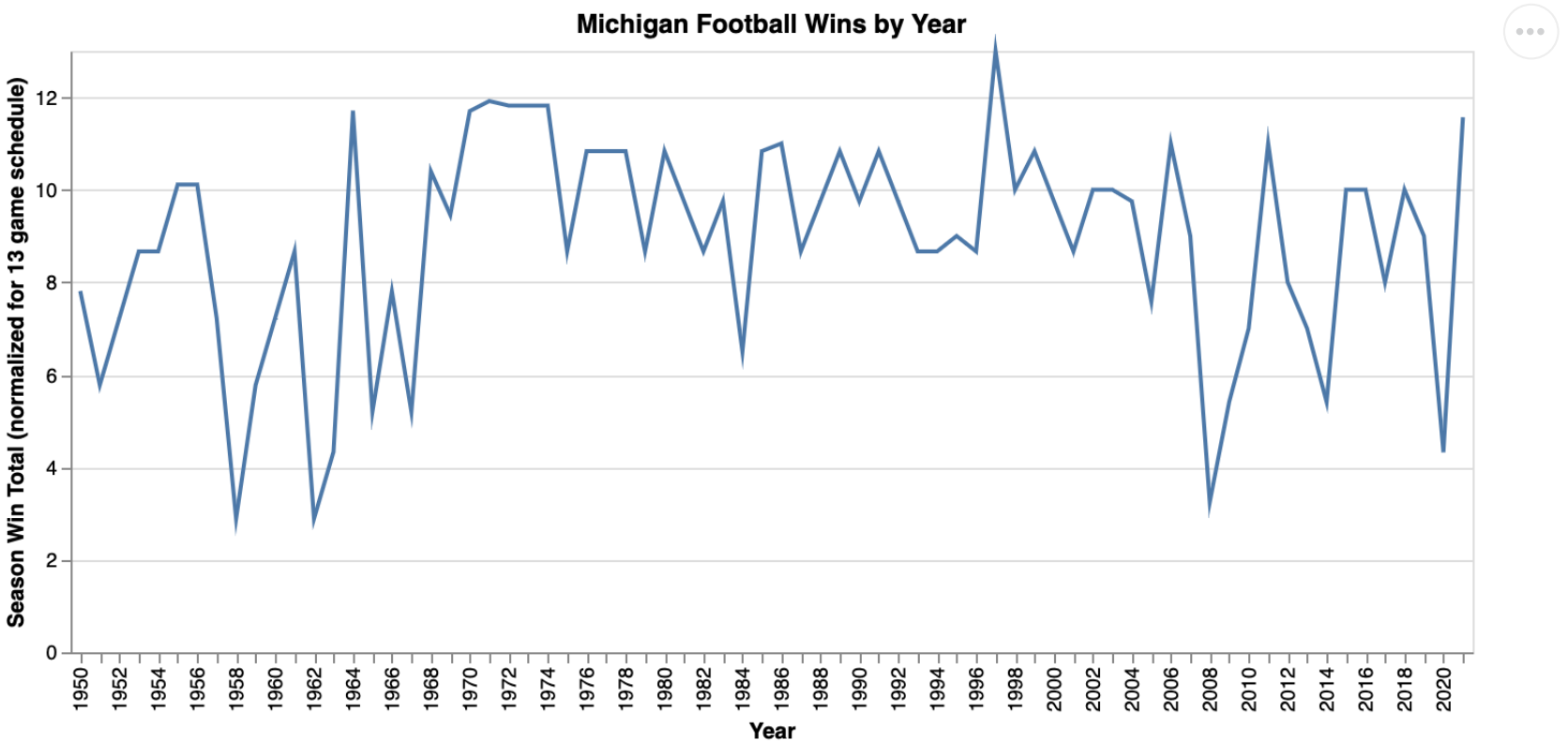
The first thing we'll do is make a very simple line graph in altair that plots the number of wins each Wolverines team had from 1950 up until the current season.

There are plenty of visualization libraries we could have leveraged to make an identical chart. For me, this chart creates more questions than answers. Remember our goal is to determine whether Jim Harbaugh is doing a good job as head football coach.

This chart doesn't help us understand things like how the team ultimately finished the season in the AP Poll ranking or even who coached that team!

In [3]:
```
alt.Chart(wins_1950).mark_line().encode(
    x = 'Year:O',
    y = alt.Y('wins_normalized:Q', title = 'Season Win Total (normalized for 13 game schedule)'),
    ).properties(width = 700, title = 'Michigan Football Wins by Year')
```

Out[3]:



Similar chart, but way more information:

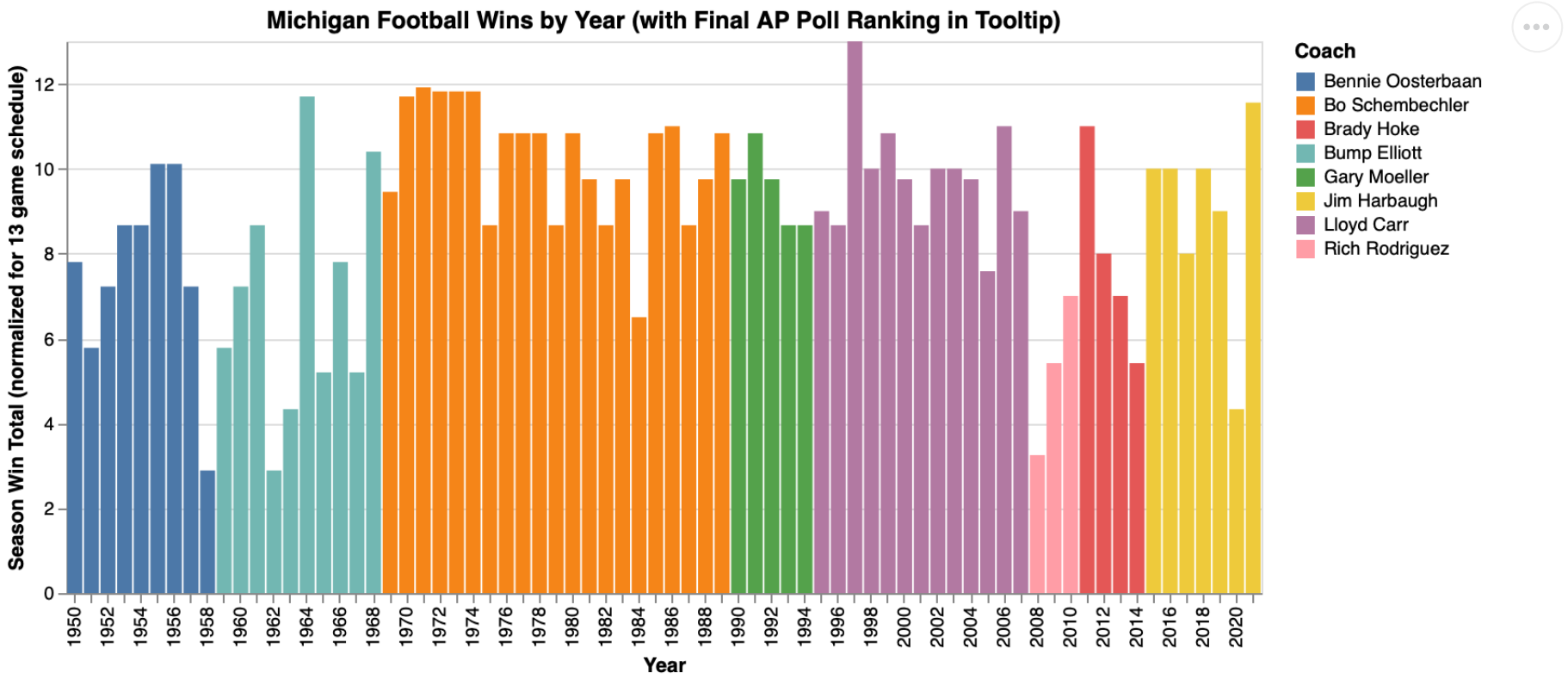
This chart takes very similar information as the line graph above and supercharges it. The first crucial improvement is that we can now see who coached each team, which allows us to see how well Coach Harbaugh stacks up to his predecessors.

However, my favorite part of this chart is that the user can hover their mouse over a bar, and see what rank the team finished in the AP Poll. This is a crucial component of our analysis, since Coach Harbaugh often gets critized for winning a lot of games, but losing key matchups against his rivals that often determine how well we fair in the final polls. This tool tip functionality is the main reason we chose altair for our analysis. (Matplotlib doens't offer a tooltip.)

Note that Coach Harbaugh routinely wins 10 or more games (just like past coaching greats like Bo Schembechler and Lloyd Carr), however he typically finishes quite a bit lower in the AP Poll rankings; largely due to his annual defeat from the team down south and underwhelming performances in Bowl Games.

In [4]:
```
alt.Chart(wins_1950).mark_bar().encode(
    x = 'Year:O',
    y = alt.Y('wins_normalized:Q', title = 'Season Win Total (normalized for 13 game schedule)'),
    color = 'Coach',
    tooltip = 'AP Post').properties(width = 650, title = 'Michigan Football Wins by Year (with Final AP
```

Out[4]:



Further Deep Dive:

Now istead of looking at the data as a time series, we're simply breaking down the number of wins each coach typically prodcued in their tenure.

Harbaugh has routinely won more games than both of predecessors Rich Rodrigues and Brady Hoke, and his median win total is only 1 fewer win than Bo Schembechler (again, this is an insight we can glean from the tooltip functionality by hovering the mouse over each boxplot).

Additionally, we can even see what year produced each coach's outliers by leverging the tooltip. The "Covid Season" wasn't kind to Coach Harbaugh: his wolverines finsihed with a 2-4 record, which equates to about 4 wins when we normalize for a 13 game schedule.

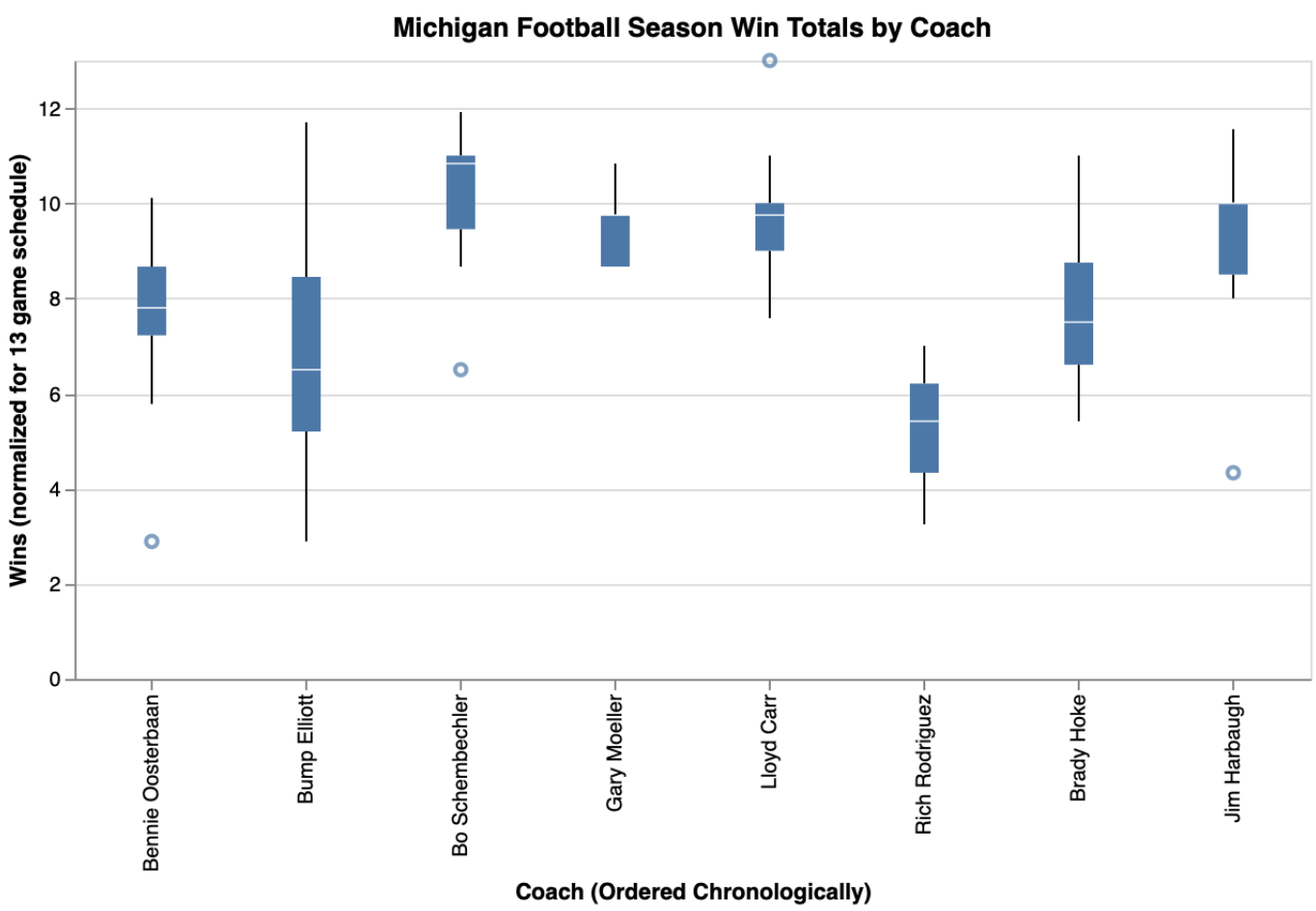
Conversely, Lloyd Carr is often regarded as Michigan's second greatest coach (behind Bo Schembechler). However, when we remove his 1997 outlier season where he won Michigan's first National Championship since World War 2, we see that he routinely won a similar number of games as Coach Harbaugh.

These last three insights are entirely thanks to the tooltip functionality, without them we woulnd't be able to explore outliers meaningully on the box plot.

In [5]:
```
sort_coaches = ['Bennie Oosterbaan','Bump Elliott', 'Bo Schembechler', 'Gary Moeller', 'Lloyd Carr', 'Ri

alt.Chart(wins_1950).mark_boxplot().encode(
    y = alt.Y('wins_normalized:Q', title = 'Wins (normalized for 13 game schedule)'),
    x = alt.X('Coach', sort = sort_coaches, title = 'Coach (Ordered Chronologically)'),
    tooltip=alt.Tooltip("Year")
).properties(width = 600, title = "Michigan Football Season Win Totals by Coach")
```

Out[5]:



More than wins: what games are Harbaugh losing that makes him controversial?

Same chart as above, but now we're looking at the Final AP Poll Rankings for each coach. This is where critcism on Harbaugh begins to become more understandable.

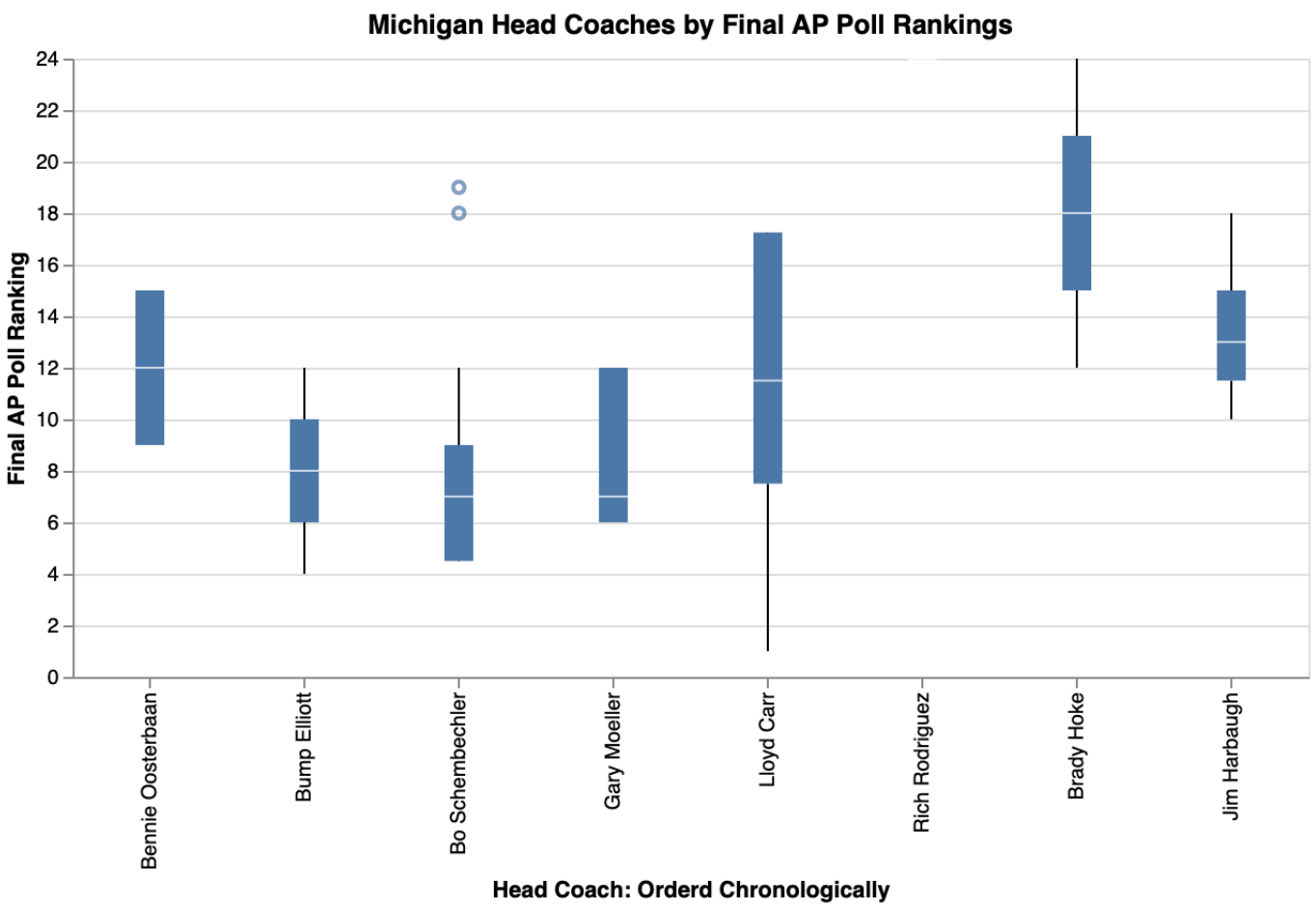
Note that Harbaugh's best season (where he finished 10th in the AP Poll) is worse than the 3rd quartile of Coach Bo's final AP Poll Rankings (which is 9th place), and similar to the median finish for Coach Carr (which is 11th place).

Again these insights are only possible thanks to Altair's tool-tip functionality.

*Note that seasons when the Wolverines fail to make the final AP Poll are excluded.

In [6]:
```
alt.Chart(wins_1950).mark_boxplot().encode(
    y = alt.Y('AP Post:Q', title = 'Final AP Poll Ranking'),
    x = alt.X('Coach', sort = sort_coaches, title = 'Head Coach: Orderd Chronologically'),
    tooltip=alt.Tooltip("Year")
).properties(width = 600, title = 'Michigan Head Coaches by Final AP Poll Rankings')
```

Out[6]:



Conclusion: Harbaugh vs the Best

Finally, we'll examine season win totals for Bo Schembechler, Lloyd Carr, and Jim Harbaugh in a stacked histogram.

This excludes the Covid Season in 2020, and the current year which obviously isn't complete yet. Harbaugh's seasons appear as an average of the two greats. He's not hitting 11 and 12 wins like Bo used to routinely acheive, and he certainly hasn't captured anything close to a National Championship like Coach Carr did in '97.

However, if we exclude the covid year, he also hasn't 'bottomed out' like Coach Carr and Schembechler. Harbaugh's worst season finsihed with just 8 wins, and if we look closer, Bo's worst season finished at an astonishing 6-6.

Michigan fans base are not comparing Harbaugh and Bo. Instead they're comparing Harbaugh with Bo of the early 1970s. Or Harbaugh with Lloyd Carr of 1997, where he won the National Championship. Bo was entitled to a 6-6 season, and more than a handful of 8-4 years as well.

Don't let Bo Schembechler's ghost haunt this program forever. Realize you have something good before it's gone - nothing's stopping us from ending up like other blue blood programs like USC or Florida State can't even manage to win 8 games in a season.

In [7]:
```
# Minor data manipulation to make quick histogram

# Remove current season, and covid season (2020)
wins_histogram = wins_1950[wins_1950['Year'] != '2021']
wins_histogram = wins_histogram[wins_histogram['Year'] != '2020']

#Pivot data out to create histogram
pivot = wins_histogram.pivot_table(values = 'wins_normalized', index = 'Year', columns = 'Coach', aggfun
pivot = pivot[['Bo Schembechler', 'Brady Hoke', 'Gary Moeller', 'Jim Harbaugh', 'Lloyd Carr']]
```

In [8]:
```
alt.Chart(pivot).transform_fold(
    #['Bo Schembechler', 'Brady Hoke', 'Gary Moeller', 'Jim Harbaugh', 'Lloyd Carr'],
    ['Bo Schembechler', 'Jim Harbaugh', 'Lloyd Carr'],
    as_=['Coach', 'wins_normalized']
).mark_area(
    opacity=0.35,
    interpolate='step'
).encode(
    alt.X('wins_normalized:Q',bin=alt.Bin(maxbins=7), title = 'Season Win Total'),
    alt.Y('count()', stack=None, scale=alt.Scale(domain=[0, 8]), title = 'Number of Season'),
    alt.Color('Coach:N')
).properties(title = "Harbaugh vs the Greats: Season Win Total", width = 375)
```

Out[8]: