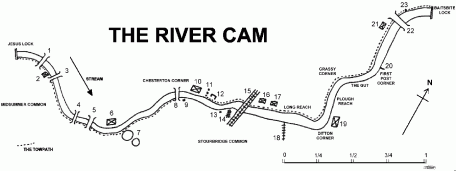
In my free time away from PhD and data science work, I (used to) enjoy rowing. Aside from obvious benefits like socialising, providing a (very intense) workout, seeing the outdoors at least a few times a week… there are really two things that I love(d) about rowing:

1. It’s the sport that is closest to a simple engineering problem. Going fast basically boils down to how in time and how hard you can get 1-8 guys to move an oar through the water. Realistically, you could probably model how good a boat of guys will row just by tracking them on a rowing machine (and I have suspicions that this is what British Rowing etc. do for national teams).
2. I learnt to row as an undergraduate at Oxford, and really got serious about it as a postgraduate student at Cambridge. This might seem like a irrelevant detail but it’s not.

**How rowing usually works**

Generally when racing boats, some n number of rowing crews line up alongside each other, and row straight down a lake (usually ~2km). The first boat to cross the finish line is generally considered the winner. For an example of such a race, see this Olympic final from 2012:

You might notice that there are four men in each boat here, each of whom are rowing. This works well on a reservoir where this race was held, but not so well on (e.g.) the River Cam that flows through Cambridge, which is both a pretty thin river, and has lots of tight corners.



the river cam

**How rowing in Cambridge/Oxford works**

Instead of these rivers, boat typically contain 8 rowers, and one cox, who is responsible for steering the boat. In lieu of the space needed to row side by side, various races across the year are run as time trials down a portion of the river. The *real* highlight of the year however, are two four day competitions in which crews line up one-behind-the-other and attempt to chase down and ‘bump’ the crew ahead (before being chased down themselves).

Upon hitting the boat that starts ahead, the two crews switch places the next day and then the race is run again, until hopefully, the positions roughly reflect the speeds of the boats.

**Predicting bumps races**

Generally therefore, if two boats line up for a bumps race, the faster one should catch the slower boat (or if the positions are reversed, the faster boat should fail to be caught by the slower boat behind). It *should* be fairly easy to predict bumps races, but it isn’t. The nature of the relative inexperience of lots of crews, the panic of the races, and the pretty tight course means mistakes are made early and often.

However, I wanted to see how possible it was. The only real data to train predictions on are the time-trial races that happen before bumps, so I’m going to see how well it’s possible to model a bumps race using the implied speeds of crews from these previous time trials.

**Libraries and Data**

For this post, I’m only going to do some simple munging and logistic regression, so I only need the (new) version of the Tidyverse (as I’m also going to play with pivot\_longer and pivot\_shorter for the [first time](http://www.robert-hickman.eu/post/cam_rowing_1/)).

library(tidyverse)

The data comes from my own scraping of race results on the river cam over the last ten years. I’ll eventually package this up properly. For now it can be found at [my Github](https://github.com/RobWHickman/CamStroker). Today I’ll just read in the raw .csv files.

#download the raw data

#wil lbe packages eventually

race\_results <- read.csv("<https://raw.githubusercontent.com/RobWHickman/CamStroker/master/data-raw/cambridge_race_results.csv>",

stringsAsFactors = FALSE)

bumps\_results <- read.csv("<https://raw.githubusercontent.com/RobWHickman/CamStroker/master/data-raw/cambridge_bumps_results.csv>",

stringsAsFactors = FALSE)

**Data Munging**

We then want to lengthen out the bumps data by days to squeeze as much data as possible out of possible combinations of boats we have data for. I need to line boats up by the start position each day, so I also init a column for this at the end

bumps\_long <- bumps\_results %>%

#pivot bumps results to longer so we can model each day of racing

pivot\_longer(., starts\_with("Day"),

names\_to = "Day", values\_to = "Bump") %>%

mutate(Bump = case\_when(

[is.na](http://is.na)(Bump) ~ 0,

TRUE ~ as.numeric(Bump)

)) %>%

group\_by(Competition, College, Year, Crew, Gender) %>%

#calculate day start and end positions

mutate(day\_end = StartPos - cumsum(Bump)) %>%

mutate(day\_start = day\_end + Bump)

head(bumps\_long)

## # A tibble: 6 x 10

## # Groups: Competition, College, Year, Crew, Gender [2]

## Competition College Year Crew Gender StartPos Day Bump day\_end

##

## 1 Lent Caius 2016 NA M 1 Day1 0 1

## 2 Lent Caius 2016 NA M 1 Day2 0 1

## 3 Lent Caius 2016 NA M 1 Day3 0 1

## 4 Lent Caius 2016 NA M 1 Day4 0 1

## 5 Lent Downing 2016 NA M 2 Day1 0 2

## 6 Lent Downing 2016 NA M 2 Day2 0 2

## # ... with 1 more variable: day\_start

And we want to do the opposite for the race data so we can efficiently join this onto the bumps data. As the speed of the crew is all we care about I calculate this as the course distance / seconds taken to get an idea of roughly how fast each crew is.

race\_wide <- race\_results %>%

#calculate implied racing speed

mutate(race\_id = paste(race, leg),

av\_speed = distance / seconds) %>%

select(Year = year, College = college, Crew = crew, Gender = gender,

race\_id, av\_speed) %>%

pivot\_wider(., id\_cols = c("Year", "College", "Crew", "Gender"),

names\_from = race\_id, values\_from = av\_speed) %>%

#rename to tidy up

rename(NSC = `Newnham Short Course NA`,

Frbrn = `Fairbairns NA`,

WH2H1 = `Winter Head 2 Head leg1`,

WH2H2 = `Winter Head 2 Head leg2`,

Rbnsn = `Robinson Head NA`) %>%

#not much data for Robinson regatta so leave out

select(-Rbnsn)

head(race\_wide)

## # A tibble: 6 x 8

## Year College Crew Gender NSC Frbrn WH2H1 WH2H2

##

## 1 2010 Catz 1 M 5.42 NA NA NA

## 2 2010 Robinson 1 M 5.41 NA NA NA

## 3 2010 Sidney 1 M 5.39 NA NA NA

## 4 2010 Caius 1 M 5.38 NA NA NA

## 5 2010 Girton 1 M 5.22 NA NA NA

## 6 2010 Churchill 1 M 5.21 NA NA NA

Then we simply join the data and calculate the implied speed differential between two crew who start a bumps race behind each other.

regression\_data <- bumps\_long %>%

ungroup() %>%

#join datasets

left\_join(race\_wide, by = c("Year", "College", "Crew", "Gender")) %>%

group\_by(Competition, Year, Gender, Day) %>%

arrange(Competition, Year, Gender, Day, day\_start) %>%

#calculate speed difference between boats starting bumps data behind each other

mutate(frbrn\_diff = Frbrn - lag(Frbrn),

NSC\_diff = NSC - lag(NSC),

WH2H1\_diff = WH2H1 - lag(WH2H1),

WH2H2\_diff = WH2H2 - lag(WH2H2)) %>%

select(-NSC, -Frbrn, -WH2H1, -WH2H2) %>%

#pivot longer for plotting

pivot\_longer(., frbrn\_diff:WH2H2\_diff,

names\_to = "race", values\_to = "speed\_difference") %>%

filter(![is.na](http://is.na)(speed\_difference)) %>%

#tidy up plotting data

filter(Bump %in% c(1, 0)) %>%

filter(Competition == "Lent")

head(regression\_data)

## # A tibble: 6 x 12

## # Groups: Competition, Year, Gender, Day [2]

## Competition College Year Crew Gender StartPos Day Bump day\_end

##

## 1 Lent Sidney 2010 2 F 40 Day1 1 39

## 2 Lent Christs 2010 2 F 41 Day1 0 41

## 3 Lent Sidney 2010 2 F 40 Day2 0 39

## 4 Lent Christs 2010 2 F 41 Day2 1 40

## 5 Lent Magdal~ 2010 2 F 46 Day2 0 45

## 6 Lent Newnham 2010 3 F 52 Day2 0 51

## # ... with 3 more variables: day\_start , race ,

## # speed\_difference

**Plotting Data**

We can then plot the data and see if the speed differential of races earlier in theyear is a useful predictor of bumping a boat ahead of you. We can model this as a logistic problem where bumping is either a 1 (to catch the boat ahead) or a 0 (did not catch). This does cut out some data in weird ways that I’ll get onto in later posts, but will do for now.

For the logistic regression I use geom\_smooth and a binomial generalised linear model. Again, there’s more we can explore here, but this is just a quick intro post so we won’t worry about standrad error etc. I also split out by Male/Female crews as I imagine gender might play a role.

Given that this week is the first race of the 2019/2020 calendar (Fairbairns) I first limited myself to only data from that race.

p1 <- regression\_data %>%

#filter only Fairbairns results

filter(race == "frbrn\_diff") %>%

ggplot(., aes(x = speed\_difference, y = Bump)) +

geom\_point() +

#model as a logistic event

geom\_smooth(method = "glm",

method.args = list(family = "binomial"),

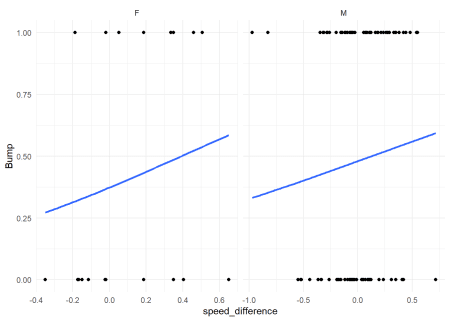
se = FALSE) +

facet\_wrap(~Gender, scales = "free\_x") +

theme\_minimal()

#plot

p1



And it seems being faster than a boat ahead of you does increase your chance of bumping, but not a huge amount.

There’s good reason to beleive Fairbairns regatta might not be the best predictor of performance later in the year. It’s the first race, where many collges are still testing out their crew. It also takes place during a weekday, so many students cannot take part, and is a 4.5km race, instead of the usual 2km of later races and bumps itself.

If we look at how all races predict later bumps success we can see much nicer logistic curves:

#do the same but for all race data

p2 <- regression\_data %>%

ggplot(., aes(x = speed\_difference, y = Bump)) +

geom\_point() +

geom\_smooth(method = "glm",

method.args = list(family = "binomial"),

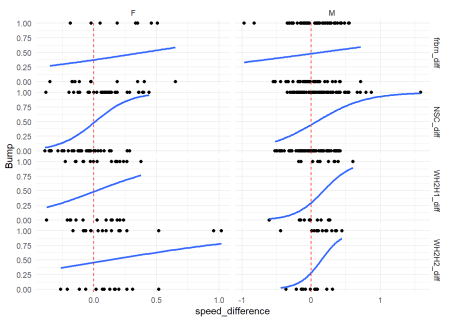
se = FALSE) +

geom\_vline(xintercept = 0, colour = "red", linetype = "dashed") +

facet\_grid(race~Gender, scales = "free\_x") +

theme\_minimal()

p2



Especially Newnham Short Course and the two legs of Winter Head 2 Head show nice curves where boats that are faster on these races have a greater chance of bumping later in the year.

There’s a lot more to do to properly model a bumps regatta, but the first step of validating our ideas and data seems to show promising results!