So you might have seen an article from someone who has a weird simpsons display picture about the [sorry state of AFL analytics](https://docs.google.com/document/d/1dCxSxHJVAeOU5tQGpLR5P0SuqlPPahb6r_pDydBclpA/edit?usp=sharing). This isn’t saying that AFL work is poor as a whole or that people within AFL circles aren’t capable of good work. This just points to a framework in place by the powers to be, that handcuff those who want to do leading work.

The main thing stopping growth in the AFL analytics community is data availability. While some might argue that the demand isn’t there, I’d like to argue that the demand is there, but the data is hard to get in a useable format and those that do get access to it don’t write in a very accessible way for the average fan who has a little bit of analytical taste.

Lets take this article on how the [Brisbane Lions are playing differently week to week](https://www.afl.com.au/news/2019-07-30/stats-files-why-the-lions-might-be-the-afls-most-versatile-team). Taking out the obvious how fans of other teams can’t get a feel for if their teams change on a week to week basis. Or the fact that league wide data isn’t available for fans to tell if there just might be a lot of variance week to week in how teams score. The data itself is presented in a poor, generally unusable format. Now that might sound harsh, so lets take a step back, its unusable for me, because I don’t like to just read numbers, I’d like to see some visualisation – which hopefully I am not in the minority on this one.

So lets take a table from the article and try to visualise it.

The table we are going to look at is the table that describes how they have scored vs different opponents and the make up of those scoring sources.

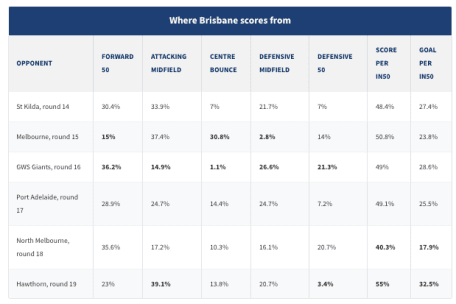


Table to recreate

What are the issues with this table as presented?

1. No visual – people I think prefer visualisation over a dump of numbers where possible
2. Some numbers have been **bolded** but there is no explanation as to why? – Leaving the reader to wonder well a lot is it to do with a significant change on the previous week? Maybe its just the max/min values of that ‘’scores from’’

So it seems like it should be an easy copy and paste job into say R and we can go right? – So thanks to [Miles](https://twitter.com/milesmcbain?lang=en) and the fantastic [datapasta](https://cran.r-project.org/web/packages/datapasta/README.html) we should be able to j ust copy and paste that data into a nice dataframe and do some visualisations.

Now unfortunetly we get this error message Could not paste clipboard as tibble. Text could not be parsed as table. Which sucks but hey we shouldn’t let that stop us because guess what, we can paste it as a vector!

library(tidyverse)

## ── Attaching packages ────────────────────────────────────────────────────────────── tidyverse 1.2.1 ──

## ✔ ggplot2 3.2.0 ✔ purrr 0.3.2

## ✔ tibble 2.1.3 ✔ dplyr 0.8.3

## ✔ tidyr 0.8.99.9000 ✔ stringr 1.4.0

## ✔ readr 1.3.1 ✔ forcats 0.4.0

## ── Conflicts ───────────────────────────────────────────────────────────────── tidyverse\_conflicts() ──

## ✖ dplyr::filter() masks stats::filter()

## ✖ dplyr::lag() masks stats::lag()

data\_set<-c("OPPONENT", NA, "FORWARD 50", NA, "ATTACKING MIDFIELD", NA, "CENTRE BOUNCE", NA, "DEFENSIVE MIDFIELD", NA, "DEFENSIVE 50", NA, "SCORE PER IN50", NA, "GOAL PER IN50", NA, "St Kilda, round 14", NA, "30.4%", NA, "33.9%", NA, "7%", NA, "21.7%", NA, "7%", NA, "48.4%", NA, "27.4%", NA, "Melbourne, round 15", NA, "15%", NA, "37.4%", NA, "30.8%", NA, "2.8%", NA, "14%", NA, "50.8%", NA, "23.8%", NA, "GWS Giants, round 16", NA, "36.2%", NA, "14.9%", NA, "1.1%", NA, "26.6%", NA, "21.3%", NA, "49%", NA, "28.6%", NA, "Port Adelaide, round 17", NA, "28.9%", NA, "24.7%", NA, "14.4%", NA, "24.7%", NA, "7.2%", NA, "49.1%", NA, "25.5%", NA, "North Melbourne, round 18", NA, "35.6%", NA, "17.2%", NA, "10.3%", NA, "16.1%", NA, "20.7%", NA, "40.3%", NA, "17.9%", NA, "Hawthorn, round 19", NA, "23%", NA, "39.1%", NA, "13.8%", NA, "20.7%", NA, "3.4%", NA, "55%", NA, "32.5%")

So the issue here is that now we have a vector that has NA in it and we also have a vector when really we want a dataframe.

So lets fix that!

First lets remove the NA

data\_fr<-data\_set[![is.na](http://is.na)(data\_set)]

Now lets make it into a data.frame

data\_fr<-matrix(data\_fr, ncol=8, byrow=TRUE)

data\_fr

## [,1] [,2] [,3]

## [1,] "OPPONENT" "FORWARD 50" "ATTACKING MIDFIELD"

## [2,] "St Kilda, round 14" "30.4%" "33.9%"

## [3,] "Melbourne, round 15" "15%" "37.4%"

## [4,] "GWS Giants, round 16" "36.2%" "14.9%"

## [5,] "Port Adelaide, round 17" "28.9%" "24.7%"

## [6,] "North Melbourne, round 18" "35.6%" "17.2%"

## [7,] "Hawthorn, round 19" "23%" "39.1%"

## [,4] [,5] [,6] [,7]

## [1,] "CENTRE BOUNCE" "DEFENSIVE MIDFIELD" "DEFENSIVE 50" "SCORE PER IN50"

## [2,] "7%" "21.7%" "7%" "48.4%"

## [3,] "30.8%" "2.8%" "14%" "50.8%"

## [4,] "1.1%" "26.6%" "21.3%" "49%"

## [5,] "14.4%" "24.7%" "7.2%" "49.1%"

## [6,] "10.3%" "16.1%" "20.7%" "40.3%"

## [7,] "13.8%" "20.7%" "3.4%" "55%"

## [,8]

## [1,] "GOAL PER IN50"

## [2,] "27.4%"

## [3,] "23.8%"

## [4,] "28.6%"

## [5,] "25.5%"

## [6,] "17.9%"

## [7,] "32.5%"

Now lets check our column names.

colnames(data\_fr)

## NULL

Whoops we still have to name our columns this comes in handy for when we want to do plotting down the track.

colnames(data\_fr) = data\_fr[1, ]

data\_fr =data\_fr[-1, ]

Now we check our dataframe and whoops I forgot to make it an actual dataframe so lets do that

data\_fr<-as.data.frame(data\_fr,stringsAsFactors = FALSE)

str(data\_fr)

## 'data.frame': 6 obs. of 8 variables:

## $ OPPONENT : chr "St Kilda, round 14" "Melbourne, round 15" "GWS Giants, round 16" "Port Adelaide, round 17" ...

## $ FORWARD 50 : chr "30.4%" "15%" "36.2%" "28.9%" ...

## $ ATTACKING MIDFIELD: chr "33.9%" "37.4%" "14.9%" "24.7%" ...

## $ CENTRE BOUNCE : chr "7%" "30.8%" "1.1%" "14.4%" ...

## $ DEFENSIVE MIDFIELD: chr "21.7%" "2.8%" "26.6%" "24.7%" ...

## $ DEFENSIVE 50 : chr "7%" "14%" "21.3%" "7.2%" ...

## $ SCORE PER IN50 : chr "48.4%" "50.8%" "49%" "49.1%" ...

## $ GOAL PER IN50 : chr "27.4%" "23.8%" "28.6%" "25.5%" ...

What you might notice here, is that we have percentages with the % sign and this forces our dataframe to have character columns when really we’d want numeric

So lets fix that, thankfully [readr](https://readr.tidyverse.org/) has a handy parse\_number function

data\_fr$`FORWARD 50`<-readr::parse\_number(data\_fr$`FORWARD 50`)

data\_fr$`ATTACKING MIDFIELD`<-readr::parse\_number(data\_fr$`ATTACKING MIDFIELD`)

Now another fun thing, you might have noticed is that when we had the variable names, that we have spaces in them. This makes it a bit annoying to work with – looking forward one of the benefits of using the tidyverse and the %>% is that we can just type the variable names, unfortunetly if we do this with variable names that have spaces this would lead to an error.

So lets fix that

names(data\_fr)<-str\_replace\_all(names(data\_fr), c(" " = "." , "," = "" ))

str(data\_fr)

## 'data.frame': 6 obs. of 8 variables:

## $ OPPONENT : chr "St Kilda, round 14" "Melbourne, round 15" "GWS Giants, round 16" "Port Adelaide, round 17" ...

## $ FORWARD.50 : num 30.4 15 36.2 28.9 35.6 23

## $ ATTACKING.MIDFIELD: num 33.9 37.4 14.9 24.7 17.2 39.1

## $ CENTRE.BOUNCE : chr "7%" "30.8%" "1.1%" "14.4%" ...

## $ DEFENSIVE.MIDFIELD: chr "21.7%" "2.8%" "26.6%" "24.7%" ...

## $ DEFENSIVE.50 : chr "7%" "14%" "21.3%" "7.2%" ...

## $ SCORE.PER.IN50 : chr "48.4%" "50.8%" "49%" "49.1%" ...

## $ GOAL.PER.IN50 : chr "27.4%" "23.8%" "28.6%" "25.5%" ...

Now lets make the rest of our variables numbers

data\_fr$CENTRE.BOUNCE<-parse\_number(data\_fr$CENTRE.BOUNCE)

data\_fr$DEFENSIVE.MIDFIELD<-parse\_number(data\_fr$DEFENSIVE.MIDFIELD)

data\_fr$DEFENSIVE.50<-parse\_number(data\_fr$DEFENSIVE.50)

data\_fr$SCORE.PER.IN50<-parse\_number(data\_fr$SCORE.PER.IN50)

data\_fr$GOAL.PER.IN50<-parse\_number(data\_fr$GOAL.PER.IN50)

str(data\_fr)

## 'data.frame': 6 obs. of 8 variables:

## $ OPPONENT : chr "St Kilda, round 14" "Melbourne, round 15" "GWS Giants, round 16" "Port Adelaide, round 17" ...

## $ FORWARD.50 : num 30.4 15 36.2 28.9 35.6 23

## $ ATTACKING.MIDFIELD: num 33.9 37.4 14.9 24.7 17.2 39.1

## $ CENTRE.BOUNCE : num 7 30.8 1.1 14.4 10.3 13.8

## $ DEFENSIVE.MIDFIELD: num 21.7 2.8 26.6 24.7 16.1 20.7

## $ DEFENSIVE.50 : num 7 14 21.3 7.2 20.7 3.4

## $ SCORE.PER.IN50 : num 48.4 50.8 49 49.1 40.3 55

## $ GOAL.PER.IN50 : num 27.4 23.8 28.6 25.5 17.9 32.5

Ok so now we have all that, what is our next issue? We need to get our data into a format that is easier to think about when plotting, to do this before we had to use functions like spread and gather which were not as intuitive as the new ones pivot\_longer() and pivot\_wider()

So how do we do that, well first we have to install the development version of [tidyr](https://tidyr.tidyverse.org/dev/articles/pivot.html) hopefully this reminds people of what they usually might be coming from (excel)

library(tidyr)

data\_fr%>%

tidyr::pivot\_longer(-OPPONENT, names\_to="scoring\_method", values\_to="pertcentage")

## # A tibble: 42 x 3

## OPPONENT scoring\_method pertcentage

##

## 1 St Kilda, round 14 FORWARD.50 30.4

## 2 St Kilda, round 14 ATTACKING.MIDFIELD 33.9

## 3 St Kilda, round 14 CENTRE.BOUNCE 7

## 4 St Kilda, round 14 DEFENSIVE.MIDFIELD 21.7

## 5 St Kilda, round 14 DEFENSIVE.50 7

## 6 St Kilda, round 14 SCORE.PER.IN50 48.4

## 7 St Kilda, round 14 GOAL.PER.IN50 27.4

## 8 Melbourne, round 15 FORWARD.50 15

## 9 Melbourne, round 15 ATTACKING.MIDFIELD 37.4

## 10 Melbourne, round 15 CENTRE.BOUNCE 30.8

## # … with 32 more rows

From there we just put it all together and try to paint hopefully a more intuitive visualisation.

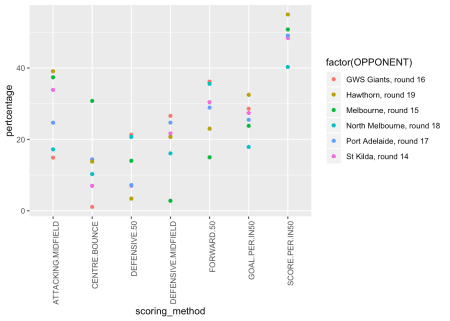
data\_fr%>%

tidyr::pivot\_longer(-OPPONENT, names\_to="scoring\_method", values\_to="pertcentage")%>%

ggplot(aes(y=pertcentage, x=scoring\_method))+

geom\_point(aes(colour=factor(OPPONENT)))+

theme(axis.text.x = element\_text(angle = 90, hjust = 1))



So what we can see here and get a feel for is the sort of values that each different scoring method had over the 6 week period. We can see that against Melbourne the lions had their highest percentage over the defined period for centre bounces and lowest from defensive midfield and forward 50. Was this a decision they made? Do other teams when playing Melbourne have the same pattern? We don’t know (data isn’t available) and Marc didn’t bother to look or it didn’t fit the narrative so was left out or looked it fit a narrative and was left out?

Some people might say that looks ok, but instead of reading top bottom, I’d rather read left to right, in other words can you flip it. Well lets do that.

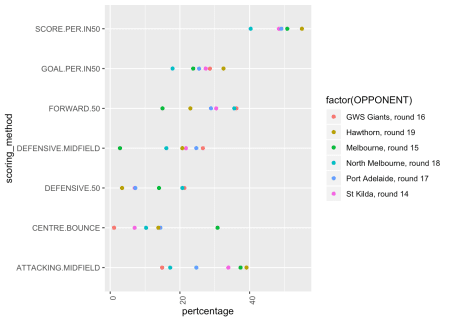
data\_fr%>%

tidyr::pivot\_longer(-OPPONENT, names\_to="scoring\_method", values\_to="pertcentage")%>%

ggplot(aes(y=pertcentage, x=scoring\_method))+

geom\_point(aes(colour=factor(OPPONENT)))+

theme(axis.text.x = element\_text(angle = 90, hjust = 1))+coord\_flip()



What else do we know about the data, well if we take out the variables SCORE.PER.IN50 and GOAL.PER.IN50 we should end up with 5 scoring methods that add up to 100.

So lets do that

data\_fr%>%

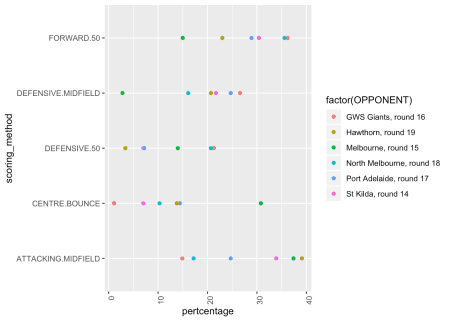
tidyr::pivot\_longer(-OPPONENT, names\_to="scoring\_method", values\_to="pertcentage")%>%

filter(!scoring\_method %in% c("SCORE.PER.IN50", "GOAL.PER.IN50") )%>%

ggplot(aes(y=pertcentage, x=scoring\_method))+

geom\_point(aes(colour=factor(OPPONENT)))+

theme(axis.text.x = element\_text(angle = 90, hjust = 1))+coord\_flip()



Now something else, you might do is that it was brought up in a talk by [Alice Sweeting](https://twitter.com/alicesweeting?lang=en) that some people are colourblind and I’m not sure about you reading, but to me these colours look pretty close to each other anyways. So lets fix that so instead of colours lets use symbols. lets also remove the legend title (yep this will require a google)

data\_fr%>%

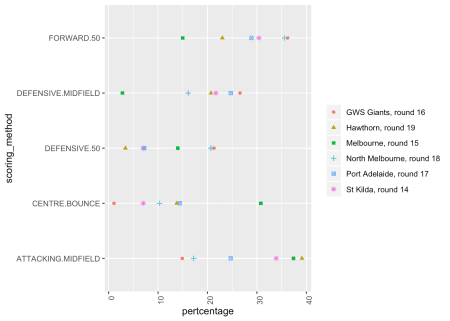
tidyr::pivot\_longer(-OPPONENT, names\_to="scoring\_method", values\_to="pertcentage")%>%

filter(!scoring\_method %in% c("SCORE.PER.IN50", "GOAL.PER.IN50") )%>%

ggplot(aes(y=pertcentage, x=scoring\_method))+

geom\_point(aes(colour=factor(OPPONENT), shape=factor(OPPONENT)))+

theme(axis.text.x = element\_text(angle = 90, hjust = 1))+coord\_flip()+theme(legend.title = element\_blank())



So there you go, thanks to wonderful people like Miles you can now copy data that you see in articles that you might want to explore further. This is particularly helpful when sites/journalists decide that instead of a visualisation you must view numbers like in an excel sheet and really who wants to do that!