[Selby Lee Steere](https://www.abc.net.au/news/2018-03-21/afl-fantasy-football-winner-kicking-charity-goals/9568554) has won [AFL fantasy](https://fantasy.afl.com.au/) two years in a row. He is the coach of [Moreiras Magic](https://twitter.com/MoreirasMagic?lang=en) and through selling his fantasy guide donates a lot of money to the [Starlight foundation](https://t.co/YSeAf2shmg).

AFL fantasy scores as opposed to supercoach scores are able to be fully derived from the [box-score statistics](https://fantasy.afl.com.au/classic/help/game-guidelines#sec-8). Why this is good it means if we wanted to, and I guess we do for the purposes of this post lets do a few things

1. Lets get the fantasy scores for all players as far back as we can go
2. Now that we have fantasy scores going back years and years, lets compare players
3. Another one of Selby fantastic tips was to think about points per minute, we can actually calculate this too!

**Step one Get the data!**

library(tidyverse)

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

## ✔ ggplot2 3.1.0 ✔ purrr 0.3.0

## ✔ tibble 2.0.1 ✔ dplyr 0.8.0.1

## ✔ tidyr 0.8.3 ✔ 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()

dataframe<-fitzRoy::get\_afltables\_stats(start\_date = "1897-01-01",end\_date = "2018-10-10")

## Returning data from 1897-01-01 to 2018-10-10

## Downloading data

##

## Finished downloading data. Processing XMLs

## Warning: Detecting old grouped\_df format, replacing `vars` attribute by

## `groups`

## Finished getting afltables data

**Step two data checks**

One of the things about AFL data is that statistics are being collected from different points in time, no one was recording the meters gained back in the 1960s.

A quick way to check when a statistic is first being collected would be to do a search or hopefully there is a data dictionary.

But in this case, lets use R and the tidyverse to do it.

library(tidyverse)

dataframe%>%

group\_by(Season)%>%

summarise(meantime=mean(Time.on.Ground..))%>%

filter(meantime>0)

## # A tibble: 16 x 2

## Season meantime

##

## 1 2003 81.8

## 2 2004 81.8

## 3 2005 81.8

## 4 2006 81.8

## 5 2007 81.8

## 6 2008 81.8

## 7 2009 81.8

## 8 2010 81.8

## 9 2011 81.8

## 10 2012 81.8

## 11 2013 81.8

## 12 2014 81.8

## 13 2015 81.8

## 14 2016 81.8

## 15 2017 81.8

## 16 2018 81.8

So what we can see from this, is that time on ground was first being collected in 2003. So if we were to work out points per minute we could only start in 2003.

**Come up with a column of fantasy scores**

We know that AFL fantasy scores are worked out as follows:

df<-data.frame(stringsAsFactors=FALSE,

Match.Stat = c("Kick", "Handball", "Mark", "Tackle", "Free Kick For",

"Free Kick Against", "Hitout", "Goal", "Behind"),

Fantasy.Points = c("3 Points", "2 Points", "3 Points", "4 Points",

"1 Point", "-3 Points", "1 Point", "6 Points",

"1 Point")

)

df

## Match.Stat Fantasy.Points

## 1 Kick 3 Points

## 2 Handball 2 Points

## 3 Mark 3 Points

## 4 Tackle 4 Points

## 5 Free Kick For 1 Point

## 6 Free Kick Against -3 Points

## 7 Hitout 1 Point

## 8 Goal 6 Points

## 9 Behind 1 Point

Now all we have to do is create a new column using the above as a formula so we can work out the fantasy scores going backwards, we can use mutate to do this.

dataframe%>%

mutate(fantasy\_score=

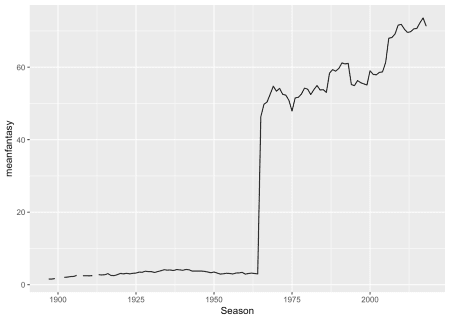
3\*Kicks +2\*Handballs+ 3\*Marks +4\*Tackles+Frees.For -

3\*Frees.Against+Hit.Outs+6\*Goals+Behinds)%>%

group\_by(Season)%>%

summarise(meanfantasy=mean(fantasy\_score))%>%

ggplot(aes(x=Season, y=meanfantasy))+geom\_line()

Looking at the graph, what we can see here is that roughly in the 60s there was a sudden jump in the statistics being collected that went on to be used to derive the AFL fantasy scores.

To find out when the statistics are first being collected we might want to do this graphically. What I am thinking what if we just had a heap of lines with each line representing one of the statistics. When it comes to plotting multiple lines on a graph its a good idea to practice the use of gather from the tidyverse

dataframe%>%

mutate(fantasy\_score=

3\*Kicks +2\*Handballs+ 3\*Marks +4\*Tackles+Frees.For -

3\*Frees.Against+Hit.Outs+6\*Goals+Behinds)%>%

group\_by(Season)%>%

summarise(meanfantasy=mean(fantasy\_score),

meankicks=mean(Kicks),

meanhandballs=mean(Handballs),

meanmarks=mean(Marks),

meantackles=mean(Tackles),

meanfreesfor=mean(Frees.For),

meansfreesagainst=mean(Frees.Against),

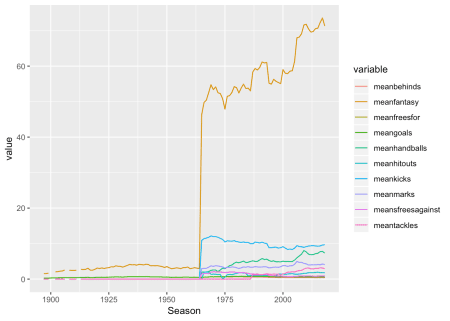
meanhitouts=mean(Hit.Outs),

meangoals=mean(Goals),

meanbehinds=mean(Behinds))%>%

gather("variable", "value",-Season) %>%

ggplot(aes(x=Season, y=value, group=variable, colour=variable))+geom\_line()



Looking at this graph, it would seem as though if we used 2000 as a cut off, that would be fine as all our data is being collected then. What we don’t want there to be an example of, is a statistic like meters gained that was only started.

Next what we want to do now that we have the fantasy scores and a rough timeline of when it makes sense that all our data is being collected lets see the average fantasy scores by player and year.

dataframe%>%

mutate(fantasy\_score=

3\*Kicks +2\*Handballs+ 3\*Marks +4\*Tackles+Frees.For -

3\*Frees.Against+Hit.Outs+6\*Goals+Behinds)%>%

group\_by(Season, First.name, Surname, ID)%>%

summarise(meanFS=mean(fantasy\_score))%>%

filter(Season>2000)

## # A tibble: 10,933 x 5

## # Groups: Season, First.name, Surname [10,879]

## Season First.name Surname ID meanFS

##

## 1 2001 Aaron Fiora 911 34.6

## 2 2001 Aaron Hamill 171 71.4

## 3 2001 Aaron Henneman 362 32.9

## 4 2001 Aaron Lord 574 48.7

## 5 2001 Aaron Shattock 1093 31.3

## 6 2001 Adam Contessa 457 49.9

## 7 2001 Adam Goodes 1012 71.5

## 8 2001 Adam Houlihan 593 54.7

## 9 2001 Adam Hunter 1072 23.3

## 10 2001 Adam Kingsley 825 66.9

## # … with 10,923 more rows

But! We know that with AFL fantasy we don’t want to include finals that’s because finals aren’t included in the competition so lets exclude finals.

dataframe%>%

mutate(fantasy\_score=

3\*Kicks +2\*Handballs+ 3\*Marks +4\*Tackles+Frees.For -

3\*Frees.Against+Hit.Outs+6\*Goals+Behinds)%>%

filter(!(Round %in% c("EF", "PF", "SF","GF","QF")))%>%

group\_by(Season, First.name, Surname, ID)%>%

summarise(meanFS=mean(fantasy\_score))%>%

filter(Season>2000)%>%arrange(desc(meanFS))

## # A tibble: 10,926 x 5

## # Groups: Season, First.name, Surname [10,872]

## Season First.name Surname ID meanFS

##

## 1 2014 Tom Rockliff 11787 135.

## 2 2012 Dane Swan 1460 134.

## 3 2018 Tom Mitchell 12196 129.

## 4 2017 Tom Mitchell 12196 127.

## 5 2012 Gary Ablett 1105 125.

## 6 2010 Dane Swan 1460 123.

## 7 2018 Jack Macrae 12166 123.

## 8 2011 Dane Swan 1460 121.

## 9 2017 Patrick Dangerfield 11700 121.

## 10 2018 Brodie Grundy 12217 120

## # … with 10,916 more rows

So what did we originally want to do, we wanted to see how Neale did in his second year to get a sense of his improvement and is that comparable to Brayshaw?

To do this we want to find the player IDS, this makes it easier to filter out the players, secondly we want to add the count of games played in season. I think the second point is important because we don’t want to be misled by a small number of games played.

dataframe%>%

mutate(fantasy\_score=

3\*Kicks +2\*Handballs+ 3\*Marks +4\*Tackles+Frees.For -

3\*Frees.Against+Hit.Outs+6\*Goals+Behinds)%>%

filter(!(Round %in% c("EF", "PF", "SF","GF","QF")))%>%

group\_by(Season, First.name, Surname, ID)%>%

summarise(meanFS=mean(fantasy\_score),

count=n())%>%

filter(Season>2000)%>%arrange(desc(Season))%>%

filter(ID %in%c("12055","12589"))

## # A tibble: 8 x 6

## # Groups: Season, First.name, Surname [8]

## Season First.name Surname ID meanFS count

##

## 1 2018 Andrew Brayshaw 12589 66.8 17

## 2 2018 Lachie Neale 12055 100. 22

## 3 2017 Lachie Neale 12055 100. 21

## 4 2016 Lachie Neale 12055 111. 22

## 5 2015 Lachie Neale 12055 102. 22

## 6 2014 Lachie Neale 12055 82.3 21

## 7 2013 Lachie Neale 12055 77.1 9

## 8 2012 Lachie Neale 12055 42.1 11

So looking at the Neale vs Brayshaw comparision, Neale actually averaged less fantasy points in his first AFL season, but he picked it up to 77.1 in his second year and a slight imporvement to 82.3 in his third.

But as Selby pointed out, he’s looking at points per minute as his metric. So lets add that in.

dataframe%>%

mutate(fantasy\_score=

3\*Kicks +2\*Handballs+ 3\*Marks +4\*Tackles+Frees.For -

3\*Frees.Against+Hit.Outs+6\*Goals+Behinds)%>%

filter(!(Round %in% c("EF", "PF", "SF","GF","QF")))%>%

group\_by(Season, First.name, Surname, ID)%>%

summarise(meanFS=mean(fantasy\_score),

meantimeonground=mean(Time.on.Ground..),

count=n())%>%

filter(Season>2000)%>%arrange(desc(Season))%>%

filter(ID %in%c("12055","12589"))

## # A tibble: 8 x 7

## # Groups: Season, First.name, Surname [8]

## Season First.name Surname ID meanFS meantimeonground count

##

## 1 2018 Andrew Brayshaw 12589 66.8 66.7 17

## 2 2018 Lachie Neale 12055 100. 80.2 22

## 3 2017 Lachie Neale 12055 100. 77.6 21

## 4 2016 Lachie Neale 12055 111. 81.6 22

## 5 2015 Lachie Neale 12055 102. 77.0 22

## 6 2014 Lachie Neale 12055 82.3 68.8 21

## 7 2013 Lachie Neale 12055 77.1 73.9 9

## 8 2012 Lachie Neale 12055 42.1 59.2 11

What we can see here, is that Brayshaw average time on ground was 66.7, Neale when he was getting just a little bit more game time (68.8 and 73.9) was getting an extra 10+ fantasy points on average. But what we can see is that his mean fantasy score and his mean time on ground is virtually one to one.

Lets see Brayshaw on a scatter plot for his debut season

dataframe%>%

mutate(fantasy\_score=

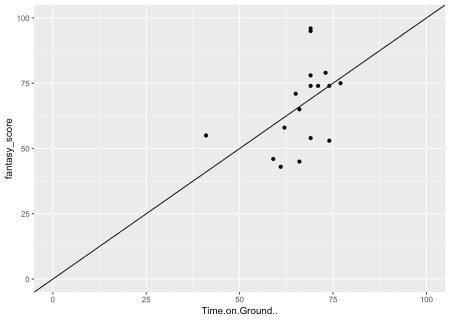
3\*Kicks +2\*Handballs+ 3\*Marks +4\*Tackles+Frees.For -

3\*Frees.Against+Hit.Outs+6\*Goals+Behinds)%>%

filter(!(Round %in% c("EF", "PF", "SF","GF","QF")))%>%

filter(ID =="12589")%>%

ggplot(aes(x=Time.on.Ground..,y=fantasy\_score))+geom\_point()+geom\_abline(intercept = 0)+ylim(0,100)+xlim(0,100)



Lets now look at Sam Powell Pepper vs Dustin Martin comparisons.

First lets find their player Ids

dataframe%>%

mutate(fantasy\_score=

3\*Kicks +2\*Handballs+ 3\*Marks +4\*Tackles+Frees.For -

3\*Frees.Against+Hit.Outs+6\*Goals+Behinds)%>%

filter(!(Round %in% c("EF", "PF", "SF","GF","QF")))%>%

filter(Surname =="Powell-Pepper")%>%select(ID)

## # A tibble: 37 x 1

## ID

##

## 1 12494

## 2 12494

## 3 12494

## 4 12494

## 5 12494

## 6 12494

## 7 12494

## 8 12494

## 9 12494

## 10 12494

## # … with 27 more rows

dataframe%>%

mutate(fantasy\_score=

3\*Kicks +2\*Handballs+ 3\*Marks +4\*Tackles+Frees.For -

3\*Frees.Against+Hit.Outs+6\*Goals+Behinds)%>%

filter(!(Round %in% c("EF", "PF", "SF","GF","QF")))%>%

filter(Surname =="Martin", First.name=="Dustin")%>%select(ID)

## # A tibble: 193 x 1

## ID

##

## 1 11794

## 2 11794

## 3 11794

## 4 11794

## 5 11794

## 6 11794

## 7 11794

## 8 11794

## 9 11794

## 10 11794

## # … with 183 more rows

Now lets do the same comparisions as we were doing for Brayshaw and Neale

dataframe%>%

mutate(fantasy\_score=

3\*Kicks +2\*Handballs+ 3\*Marks +4\*Tackles+Frees.For -

3\*Frees.Against+Hit.Outs+6\*Goals+Behinds)%>%

filter(!(Round %in% c("EF", "PF", "SF","GF","QF")))%>%

group\_by(Season, First.name, Surname, ID)%>%

summarise(meanFS=mean(fantasy\_score),

meantimeonground=mean(Time.on.Ground..),

count=n())%>%

filter(Season>2000)%>%arrange(desc(Season))%>%

filter(ID %in%c("11794","12494"))

## # A tibble: 11 x 7

## # Groups: Season, First.name, Surname [11]

## Season First.name Surname ID meanFS meantimeonground count

##

## 1 2018 Dustin Martin 11794 92.4 82.9 21

## 2 2018 Sam Powell-Pepper 12494 75 74.6 16

## 3 2017 Dustin Martin 11794 114. 85 22

## 4 2017 Sam Powell-Pepper 12494 69.9 68 21

## 5 2016 Dustin Martin 11794 107. 83.1 22

## 6 2015 Dustin Martin 11794 104. 79.8 22

## 7 2014 Dustin Martin 11794 97.6 82.9 21

## 8 2013 Dustin Martin 11794 97.0 81.8 22

## 9 2012 Dustin Martin 11794 84.8 80.4 20

## 10 2011 Dustin Martin 11794 89.5 84.4 22

## 11 2010 Dustin Martin 11794 71.5 78.2 21

Lets now look at a similar scatter plot.

dataframe%>%

mutate(fantasy\_score=

3\*Kicks +2\*Handballs+ 3\*Marks +4\*Tackles+Frees.For -

3\*Frees.Against+Hit.Outs+6\*Goals+Behinds)%>%

filter(!(Round %in% c("EF", "PF", "SF","GF","QF")))%>%

group\_by(Season, First.name, Surname, ID)%>%

summarise(meanFS=mean(fantasy\_score),

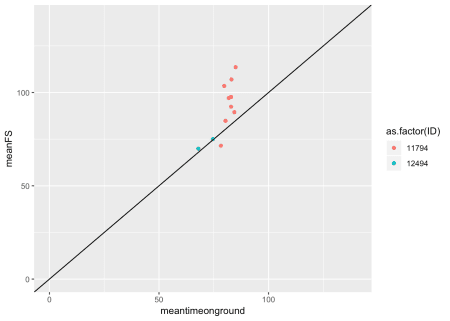
meantimeonground=mean(Time.on.Ground..),

count=n())%>%

filter(Season>2000)%>%arrange(desc(Season))%>%

filter(ID %in%c("11794","12494"))%>%

ggplot(aes(x=meantimeonground, y=meanFS))+geom\_point(aes(colour=as.factor(ID)))+geom\_abline(intercept = 0)+ylim(0,140)+xlim(0,140)



So hopefully now what you have is a rough framework in place so you can easily filter out players you think are similar, track their careers and use 2 time afl fantasy winners tips by looking at time on ground as well!