

Data Visualization

This is my attempt at conveying line performance to farmers, and could also be applicable to a manuscript. Since I am not sure how these experiments are conducted I am only displaying the difference data points since there would be too much data to display and clutter the graphs. The “raw” data could be displayed in a supplemental table. First step is to transform the data to be positive and negative in relation to ZZ Madsen, not colour coded and impute any missing data

Radial plots

Also called radar and spider plots, they have a self explanatory name and may be useful for displaying the data but I’m not sure it’s very clear to people that haven’t seen them before. For the purpose of this data only the points should be read and not the lines or area of a line.

PROs:

- * Eye catching
- * Easy to read once you know how
- * Values are converted to percentages (could also be a negative)

CONS:

- * Not sure significance can be displayed
- * Values are converted to percentages
- * Can get cluttered in the centre
- * Difficulty in reading, read the points, not the lines
- * Data can look biased based on the trait order around the diagram
- * Multiple graphs are potentially needed (as shown by the use of three graphs) to avoid cluttering figures with too many data points
- * Cannot seem to handle missing data (had to use zero for missing), possibly an easy fix

```
Club_Wheat_Radial =  
read.table("C:\\Users\\shaun.clare\\Downloads\\Club_Wheat_Radial.txt", header  
= TRUE, sep = "", row.names="VARIETY")  
  
club_wheat_solvent <- Club_Wheat_Radial %>%  
  as_tibble(row.names = "VARIETY") %>%  
  mutate_at(vars(-VARIETY), rescale) %>%  
  tail(7) %>%  
  select(1, 16:20)  
  
plot1 <- ggadar(club_wheat_solvent, axis.label.size = 4, group.line.width =  
1, group.point.size = 2, grid.label.size = 4) + labs(caption='solvent  
traits') + labs(caption = "Data Source: Kim Campbell | Plot generated by  
Shaun Clare")
```

```

club_wheat_kernel <- Club_Wheat_Radial %>%
  as_tibble(rownames = "VARIETY") %>%
  mutate_at(vars(-VARIETY), rescale) %>%
  tail(7) %>%
  select(1:2,4:5,8,10)

plot2 <- ggadar(club_wheat_kernel, axis.label.size = 4, group.line.width =
1, group.point.size = 2, grid.label.size = 4) + labs(caption='kernel
traits') + labs(caption = "Data Source: Kim Campbell | Plot generated by
Shaun Clare")

club_wheat_millingproduct <- Club_Wheat_Radial %>%
  as_tibble(rownames = "VARIETY") %>%
  mutate_at(vars(-VARIETY), rescale) %>%
  tail(7) %>%
  select(1,6:7,9,14:15)

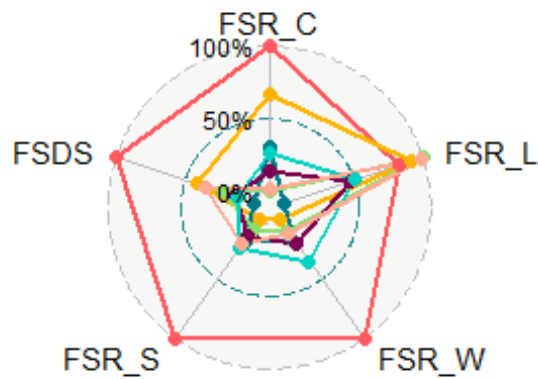
plot3 <- ggadar(club_wheat_millingproduct, axis.label.size = 4,
group.line.width = 1, group.point.size = 2, grid.label.size = 4) +
labs(caption='milling/product traits') + labs(caption = "Data Source: Kim
Campbell | Plot generated by Shaun Clare")

club_wheat_all <- Club_Wheat_Radial %>%
  as_tibble(rownames = "VARIETY") %>%
  mutate_at(vars(-VARIETY), rescale) %>%
  tail(7) %>%
  select(1:20)

plot4 <- ggadar(club_wheat_all, axis.label.size = 4, group.line.width = 1,
group.point.size = 2, grid.label.size = 4) + labs(caption='milling/product
traits') + labs(caption = "Data Source: Kim Campbell | Plot generated by
Shaun Clare")
plot1

```

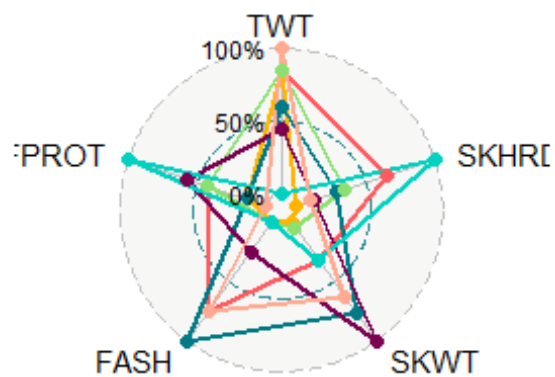
- Daws
- Lewjain
- Orcf102
- Puma
- Stephens
- Tubbs06



Source: Kim Campbell | Plot generated by Shaun Clare

plot2

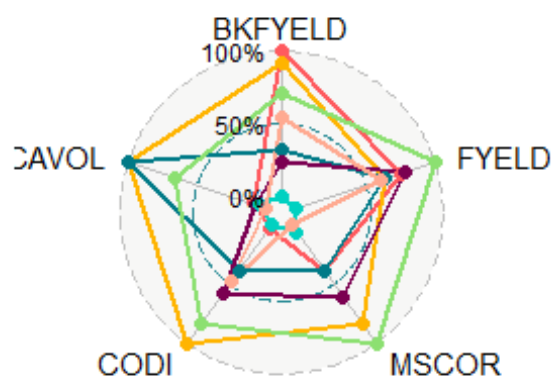
- Daws
- Lewjain
- Orcf102
- Puma
- Stephens
- Tubbs06



Source: Kim Campbell | Plot generated by Shaun Clare

plot3

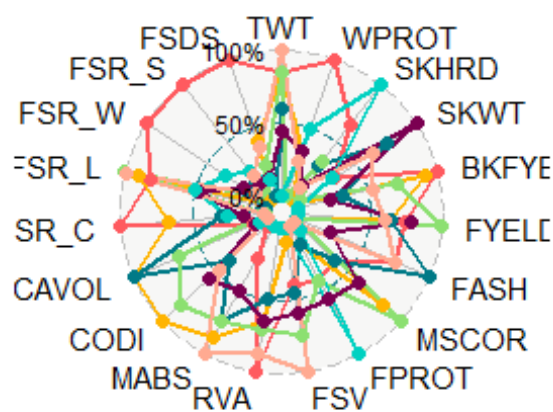
- Daws
- Lewjain
- Orcf102
- Puma
- Stephens
- Tubbs06



Source: Kim Campbell | Plot generated by Shaun Clare

plot4

- Daws
- Lewjain
- Orcf102
- Puma
- Stephens
- Tubbs06



Source: Kim Campbell | Plot generated by Shaun Clare

```
#grid.arrange(plot1, plot2, plot3, ncol=3)
#
#use theme(legend.position = 'none') + for removing legend
```

Dumbell plots

Dumbell plots are named as they typically use two data points that are joined to form a dumbbell (typically use the ggalt package but can only handle two data points). I found this work around but am still working on aesthetics and trimming code down

PROs:

- * The ZZ Madsen check is at the centre point, making it very easy to see how other lines differ
- * Very concise visual for all traits at once
- * Everyone should be to understand the graph intuitively or with quick explanation

CONS:

- * Not sure if significance can be displayed
- * Can get cluttered if many lines have a similar values (maybe add labels?)
- * Traits that have a small scale could be swamped by traits with larger scales (could be fixed with a log scale)
- * Cannot find a good colour scheme (should be an easy fix)

```
traits = c("TWT", "WPROT", "SKHRD", "SKWT", "BKFYELD", "FYELD", "FASH",
"MSCOR", "FPROT", "FSV", "RVA", "MABS", "CODI", "CAVOL", "FSR_C", "FSR_L",
"FSR_W", "FSR_S", "FSDS")

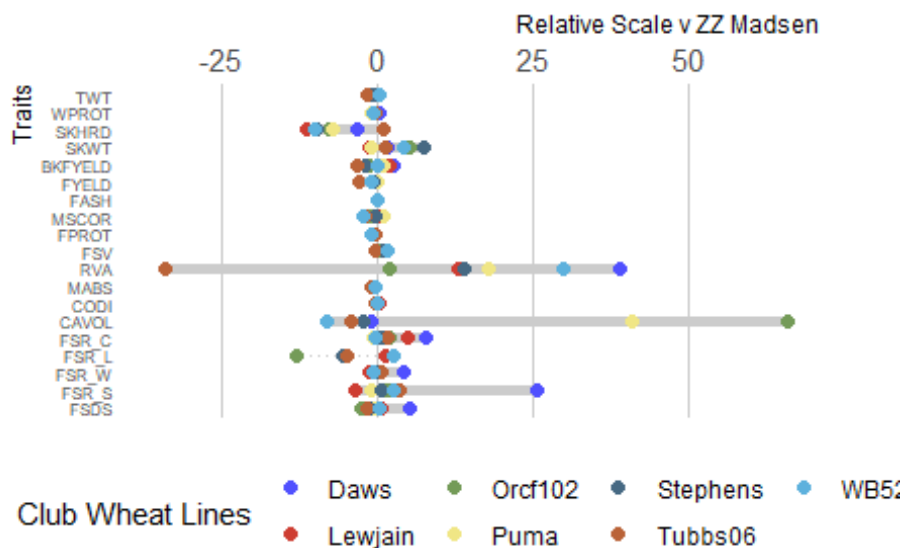
tibble(
  Daws = c(0.20, 0.50, -3.10, 1.60, 2.70, -0.60, 0.01, -0.80, -
0.50, 0.60, 39.00, -0.80, -0.09, -1.00, 7.80, NA, 4.40, 25.90,
5.40),
  Lewjain = c(0.20, -0.30, -11.20, -1.20, 2.20, -0.90, -0.02, 0.60, -
0.70, 0.00, 13.00, -0.30, 0.32, 66.00, 5.10, 1.30, -1.10, -3.30,
0.80),
  Orcf102 = c(-0.30, -0.70, -7.60, 5.40, -1.30, -0.90, 0.02, -
0.80, -0.70, 0.70, 2.00, -0.40, 0.06, 66.00, 2.10, -12.70, -0.10,
1.90, -2.40),
  Puma = c(0.20, -0.70, -7.00, -0.80, 1.00, 0.10, -0.02, 1.10, -
0.50, 1.30, 18.00, -0.40, 0.25, 41.00, -0.40, 2.70, -0.60, -1.00,
-0.60),
  Stephens = c(-0.60, -0.30, -9.70, 7.50, -1.80, -0.50, -0.01, -
0.10, -0.40, 1.00, 14.00, -0.60, 0.14, -2.00, 0.80, -5.30, 0.00,
0.70, -1.30),
  Tubbs06 = c(-1.50, -0.10, 1.20, 1.50, -3.20, -2.70, -0.02, -
1.80, -0.10, -0.20, -34.00, -1.00, -0.10, -4.00, 1.80, -4.90, 0.80,
3.70, -1.40),
  WB528 = c(0.50, -0.40, -10.00, 4.30, 0.00, -1.00, 0.01, -2.00, -
0.80, 1.80, 30.00, -0.20, 0.10, -8.00, -0.30, 2.60, -0.50, 2.70,
0.40),
```

```

cat = factor(traits[1:19], levels = rev(traits[1:19])) -> xdf

ggplot() +
  geom_segment(data = gather(xdf, measure, val, -cat) %>%
    group_by(cat) %>%
    top_n(-1) %>%
    slice(1) %>%
    ungroup(),
    aes(x = 0, xend = val, y = cat, yend = cat),
    linetype = "dotted", size = 0.5, color = "gray80") +
  geom_segment(data = gather(xdf, measure, val, -cat) %>%
    group_by(cat) %>%
    summarise(start = range(val)[1], end = range(val)[2]) %>%
    ungroup(),
    aes(x = start, xend = end, y = cat, yend = cat),
    color = "gray80", size = 2) +
  geom_point(
    data = gather(xdf, measure, value, -cat),
    aes(value, cat, color = measure),
    size = 2) +
  scale_x_comma(position = "top", limits = c(-40, 70)) +
  scale_color_igv(name = "Club Wheat Lines") +
  labs(x = "Relative Scale v ZZ Madsen", y = "Traits", title = NULL, caption
= "Data Source: Kim Campbell | Plot generated by Shaun Clare") +
  theme_ipsum_rc(grid = "X") +
  theme(legend.position = "bottom", axis.text.y = element_text(size=6))

```



Data Source: Kim Campbell | Plot generated by Shaun Clare