Candidacy Proposal for Handling Ties Using Parallel Coordinate Plots in the ggpcp Package (R)

Denise Bradford

1. Introduction

This proposal outlines a systematic approach to visually distinguish tied numerical values in multidimensional datasets by employing parallel coordinate plots (PCPs). Parallel coordinates, first popularized by Alfred Inselberg, are a powerful technique for investigating patterns across multiple attributes simultaneously (Inselberg 2009). However, when datasets contain exact numerical ties, the resulting overlapping lines in PCPs can obscure critical distinctions. To address this, we propose a method for introducing controlled spacing to tied values. This functionality will be integrated into the **ggpcp** package in R, ensuring a streamlined workflow for users seeking enhanced clarity in their parallel coordinate visualizations.

Importantly, our approach complements recent work on **generalized parallel coordinate plots (GPCPs)**—an extension of PCPs that supports categorical variables (VanderPlas et al. 2023). The *ggpcp* package for R implements these GPCPs using a *grammar of graphics* framework, which seamlessly incorporates both continuous and categorical variables in a single parallel coordinate plot. One of the key contributions of that work is a robust tie-breaking mechanism for categorical variables, ensuring that individual observations can be traced across multiple dimensions, even when categories induce identical or "tied" values. By adding a numerical tie-breaking technique for continuous data as proposed here, we further refine GPCPs' capacity to handle the visualization of real-world datasets exhibiting many types of ties.

---

2. Background and Motivation

2.1 Parallel Coordinate Plots

Parallel coordinate plots assign each dimension of an n-dimensional dataset to a vertical axis arranged in parallel (Wegman 1990). Each observation is drawn as a polyline connecting its values on these axes, providing a visual representation that can illuminate underlying data structures.

2.2 Numerical Ties and Visual Overlap

When multiple observations share the same value in a given dimension, their polylines perfectly overlap, creating "visual collisions." This masks information about distribution, density, or potential outliers. Introducing a small offset ("jitter") to these tied values can mitigate overlap without distorting the overall relationships in the data (Peng, Ward, and Rundensteiner 2004). In the context of generalized parallel coordinate plots, careful tie-handling is equally essential for both continuous and categorical variables (VanderPlas et al. 2023).

---

3. Proposed Method for Tie Spacing in `ggpcp`

3.1 Overview

To handle overlapping polylines caused by numerical ties, we propose integrating **two R functions** into the ggpcp package. Both functions introduce minimal, controlled spacing before the parallel coordinates are drawn. Users can select either a straightforward, fixed tie-band approach (**Solution #1**) or a more adaptive method based on data properties (**Solution #2**). These solutions dovetail with ggpcp's built-in handling of categorical ties (VanderPlas et al. 2023), ensuring that *all* forms of ties—categorical or continuous—are resolved before rendering the PCP.

Below are the core functions that implement these approaches:

```
# Solution #1
numerical_tie_breaker_solution1 <- function(values, tie_band = 0.05) {
  # Sort the values
  sorted_values <- sort(values)

  # Initialize the adjusted values
  adjusted_values <- sorted_values

  # Loop through the values to add tie band for ties
  for (i in seq_along(sorted_values)) {
    if (i > 1 && sorted_values[i] == sorted_values[i - 1]) {
      adjusted_values[i] <- adjusted_values[i - 1] + tie_band
    }
  }

  # Return the adjusted values
  return(adjusted_values)
}

# Solution #2: Dynamic tie band based on data properties
auto_fraction <- function(values, scale_factor = 0.05) {
  unique_vals <- unique(sort(values))
  diffs <- diff(unique_vals)

  # Keep only positive gaps
```

```r
  positive_diffs <- diffs[diffs > 0]

  # If everything is identical or only one unique value, return a small default
  if (length(positive_diffs) == 0) {
    return(scale_factor)
  }

  # A robust choice: median of these positive gaps
  typical_gap <- median(positive_diffs)

  # Return a fraction that is a multiple of the median gap
  scale_factor * typical_gap
}

numerical_tie_breaker_solution2 <- function(values, base_fraction = 0.05) {
  # 1) Determine if all values are whole numbers
  is_whole_number <- all(abs(values - round(values)) < .Machine$double.eps^0.5)

  # 2) If data are all whole numbers, use a small fraction increment
  if (is_whole_number) {
    tie_band <- base_fraction

  } else {
    # For continuous data, attempt a normality check using Shapiro-Wilk if n >= 3
    is_normal <- FALSE
    if (length(values) >= 3) {
      shapiro_p <- shapiro.test(values)$p.value
      is_normal <- (shapiro_p > 0.05)
    }

    # Check for outliers using 1.5 * IQR rule
    q <- stats::quantile(values, probs = c(0.25, 0.75))
    iqr_value <- q[2] - q[1]
    lower_bound <- q[1] - 1.5 * iqr_value
    upper_bound <- q[2] + 1.5 * iqr_value
    has_outliers <- any(values < lower_bound) || any(values > upper_bound)

    # Decide tie_band logic
    # If data appear normal and have no outliers, use fraction * SD; otherwise use fraction
    if (is_normal && !has_outliers) {
      tie_band <- base_fraction * stats::sd(values)
    } else {
      tie_band <- base_fraction * iqr_value
    }
  }

  # 3) Sort the values
  sorted_values <- sort(values)

  # 4) Adjust for ties by adding tie_band to each duplicate
  adjusted_values <- sorted_values
```

```
  for (i in seq_along(sorted_values)) {
    if (i > 1 && sorted_values[i] == sorted_values[i - 1]) {
      adjusted_values[i] <- adjusted_values[i - 1] + tie_band
    }
  }

  return(adjusted_values)
}
```

Key Features of Each Approach

1. **Solution #1** – *Static Tie Band*
    - Relies on a user-defined `tie_band` to offset tied values.
    - Offers simplicity and predictable increments in spacing.
2. **Solution #2** – *Dynamic Tie Band*
    - Adjusts tie spacing based on data properties (e.g., normality checks, IQR, and outlier detection).
    - More adaptive, especially useful for heterogeneous or continuous data with varying scales.

3.2 Integration into `ggpcp`

When integrated into ggpcp, users can preprocess their numeric columns with either function prior to plotting. The parallel coordinate plot (via `geom_parallel()` or similar functionality in ggpcp) then operates on the adjusted (tie-spaced) data, preventing line overlap. This mirrors how ggpcp already handles ties in categorical data (VanderPlas et al. 2023), thereby providing a unified tie-breaking solution for both numeric and factor variables.

For instance:

```
library(ggpcp)
library(tidyverse)
```

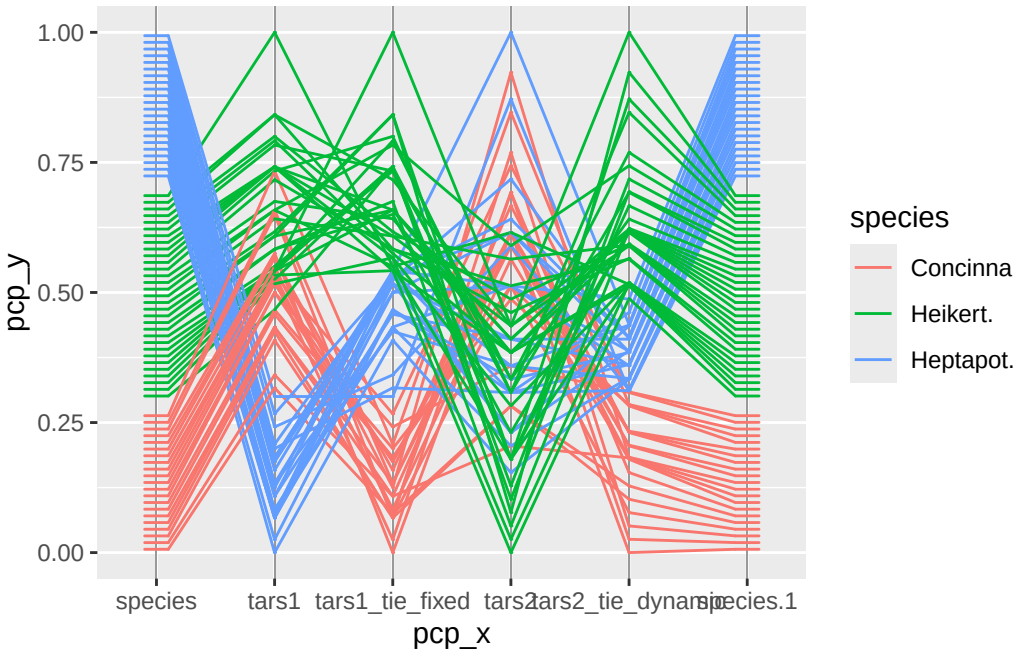
```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.1     v tibble    3.2.1
v lubridate 1.9.3     v tidyr     1.3.1
v purrr     1.0.2
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to beco
```

```r
data(flea, package = "GGally")
# Example usage

# Apply static tie spacing to column 'tars1' only
flea$tars1_tie_fixed <- numerical_tie_breaker_solution1(flea$tars1, tie_band = 0.01)

# Apply dynamic tie spacing to column 'tars2'
flea$tars2_tie_dynamic <- numerical_tie_breaker_solution2(flea$tars2)

flea %>%
  pcp_select(species, c(2,8,3, 9), species) %>%
  pcp_scale(method="uniminmax") %>%
  pcp_arrange() %>%
  ggplot(aes_pcp()) +
    geom_pcp_axes() +
    geom_pcp(aes(colour = species))
```



---

4. Implementation Roadmap

1. **Data Preparation**
   - Ensure data are in a tidy format compatible with ggpcp.
   - Normalize or standardize dimensions if necessary, so the offset remains meaningful and does not distort the scale.
2. **Integration into ggpcp**
   - Incorporate both tie-breaker functions as user-facing options within ggpcp.

- Expose parameters (e.g., `tie_band`, `base_fraction`) through well-documented function arguments and vignettes.

3. **Rendering Parallel Coordinates**
   - Use ggpcp's parallel coordinate functions on the tie-spaced data.
   - Provide relevant axis annotations, legends, and tooltips for interactive capabilities.
4. **Testing and Validation**
   - Assess the impact of tie spacing by comparing plots with and without offsets.
   - Gather feedback from early adopters to fine-tune default parameters and ensure user-friendliness.

---

5. Historical Context of Multidimensional Data Visualization

5.1 Early Developments

The concept of parallel coordinates dates back to Maurice d'Ocagne's work on "coordinate parallelism" (d'Ocagne 1885). While his contributions did not see immediate mainstream adoption, they formed the conceptual foundation for later multidimensional plotting.

5.2 Alfred Inselberg's Contributions

Alfred Inselberg developed the mathematical underpinnings of parallel coordinates for analyzing high-dimensional data, demonstrating the approach's strength in various domains (Inselberg 1985, 2009).

5.3 Wegman's Statistical Insights

Wegman (1990) further popularized parallel coordinates by integrating statistical perspectives, showing how PCPs can be leveraged in exploratory data analysis. Subsequent advances in interactive visualization have enabled real-time manipulation of axes, brushing, and filtering, allowing users to distill meaningful insights from large datasets.

5.4 Generalized Parallel Coordinates

More recently, VanderPlas et al. (2023) extended PCPs to handle both categorical and continuous variables within a single framework, implemented via the R package ggpcp. Their work, *Penguins Go Parallel*, details a grammar-of-graphics approach that accommodates a mix of variable types by applying tie-breaking and ordering strategies on categorical data to preserve the continuity of observation trajectories. Our proposed numerical tie-breaker further complements ggpcp's existing functionalities by addressing ties among continuous variables, ensuring that PCPs and GPCPs can highlight subtle distinctions within both categorical and numeric domains.

---

6. Expected Outcomes

1. **Improved Visual Clarity**
   - The introduction of tie spacing in ggpcp will ensure that overlaid lines become distinguishable, unveiling patterns and anomalies otherwise hidden by overlap.
2. **Enhanced Data Interpretation**
   - Researchers and data analysts will be able to more accurately interpret subtle differences in high-dimensional datasets, especially in domains like bioinformatics, finance, and engineering.
3. **Comprehensive Tie-Handling**
   - By integrating both numeric and categorical tie-breaking, ggpcp users enjoy a complete solution for parallel coordinate visualization, building upon recent advancements in generalized parallel coordinate plots (VanderPlas et al. 2023).
4. **Easy Integration with R Ecosystem**
   - Users of ggpcp will have a ready-to-use, integrated solution for tie handling without resorting to external data preprocessing or manual jittering methods.

---

7. Conclusion

Parallel coordinates have evolved from early geometric solutions (d'Ocagne 1885) to a robust, interactive technique for visualizing high-dimensional data (Inselberg 1985; Wegman 1990). However, numerical ties remain a persistent challenge, causing overlapping lines that diminish interpretability. By integrating minimal offset mechanisms (both static and dynamic) directly into the ggpcp package, this proposal ensures a streamlined approach that balances the preservation of data fidelity with the need for visual clarity. In tandem with recent developments in generalized parallel coordinate plots (VanderPlas et al. 2023), the resulting enhancement will aid analysts in effectively discerning underlying structures and relationships across multiple dimensions.

References

d'Ocagne, Maurice. 1885. "Coordonnées parallèles et axiales : Méthode de transformation géométrique et procédé nouveau de calcul graphique déduits de la considération des coordonnées parallèles." *Gauthier-Villars*, 112. https://archive.org/details/coordonnesparal00ocaggoog/page/n10.

Inselberg, Alfred. 1985. "The plane with parallel coordinates." *The Visual Computer* 1 (2): 69–91. https://doi.org/10.1007/BF01898350.

———. 2009. "Parallel Coordinates: Interactive Visualisation for High Dimensions." *Trends in Interactive Visualization: State-of-the-Art Survey*, 49–78.

Peng, Wei, Matthew O Ward, and Elke A Rundensteiner. 2004. "Clutter Reduction in Multi-Dimensional Data Visualization Using Dimension Reordering." In *IEEE Symposium on Information Visualization*, 89–96. IEEE.

VanderPlas, Susan, Yawei Ge, Antony Unwin, and Heike Hofmann. 2023. "Penguins Go Parallel: A Grammar of Graphics Framework for Generalized Parallel Coordinate Plots." *Journal of Computational and Graphical Statistics* 32 (4): 1572–87.

Wegman, Edward J. 1990. "Hyperdimensional data analysis using parallel coordinates." *Journal of the American Statistical Assoiation* 85: 664–75.