

8. ggforce

Zooming, Annotationen und Gruppenmarkierung

Dr. Paul Schmidt

```
for (pkg in c("tidyverse", "gapminder", "ggforce")) {
  if (!require(pkg, character.only = TRUE)) install.packages(pkg)
}
```

Das Paket {ggforce} von Thomas Lin Pedersen erweitert ggplot2 um spezialisierte Geoms und Facets, die sich mit Bordmitteln nur schwer umsetzen lassen. Waehrend ggplot2 den Grossteil alltaeglicher Visualisierungsaufgaben abdeckt, erfordern bestimmte Darstellungen - etwa das Hineinzoomen in einen ueberfuellten Plotbereich oder das Zeichnen von Formen um Punktgruppen - dedizierte Werkzeuge.

Dieses Kapitel konzentriert sich auf drei besonders nuetzliche Features von ggforce: das Zoomen in Plotbereiche mit `facet_zoom()`, das Markieren von Gruppen mit statistischen Ellipsen ueber `geom_mark_ellipse()` und das Umranden von Gruppen mit konvexen Huellen mittels `geom_mark_hull()`. Diese Werkzeuge sind besonders wertvoll fuer explorative Analysen und Praesentationen, bei denen man die Aufmerksamkeit auf bestimmte Cluster oder Teilmengen der Daten lenken moechte.

Fuer die Zoom-Beispiele verwenden wir den gapminder-Datensatz (gefiltert auf 2007) und fuer die Gruppenmarkierung den klassischen iris-Datensatz, da seine drei gut getrennten Arten die visuellen Unterschiede besonders deutlich machen.

facet_zoom

Streudiagramme realer Daten enthalten oft Bereiche, in denen sich viele Punkte ueberlagern, waehrend andere Bereiche duenn besiedelt sind. Ein klassisches Beispiel ist das BIP pro Kopf: Eine Handvoll wohlhabender Laender streckt die x-Achse so weit, dass die Unterschiede zwischen einkommensschwachen Laendern unsichtbar werden. `facet_zoom()` loest dieses Problem, indem sowohl der vollstaendige Plot als auch ein vergroesserter Ausschnitt nebeneinander angezeigt werden.

Basis-Plot

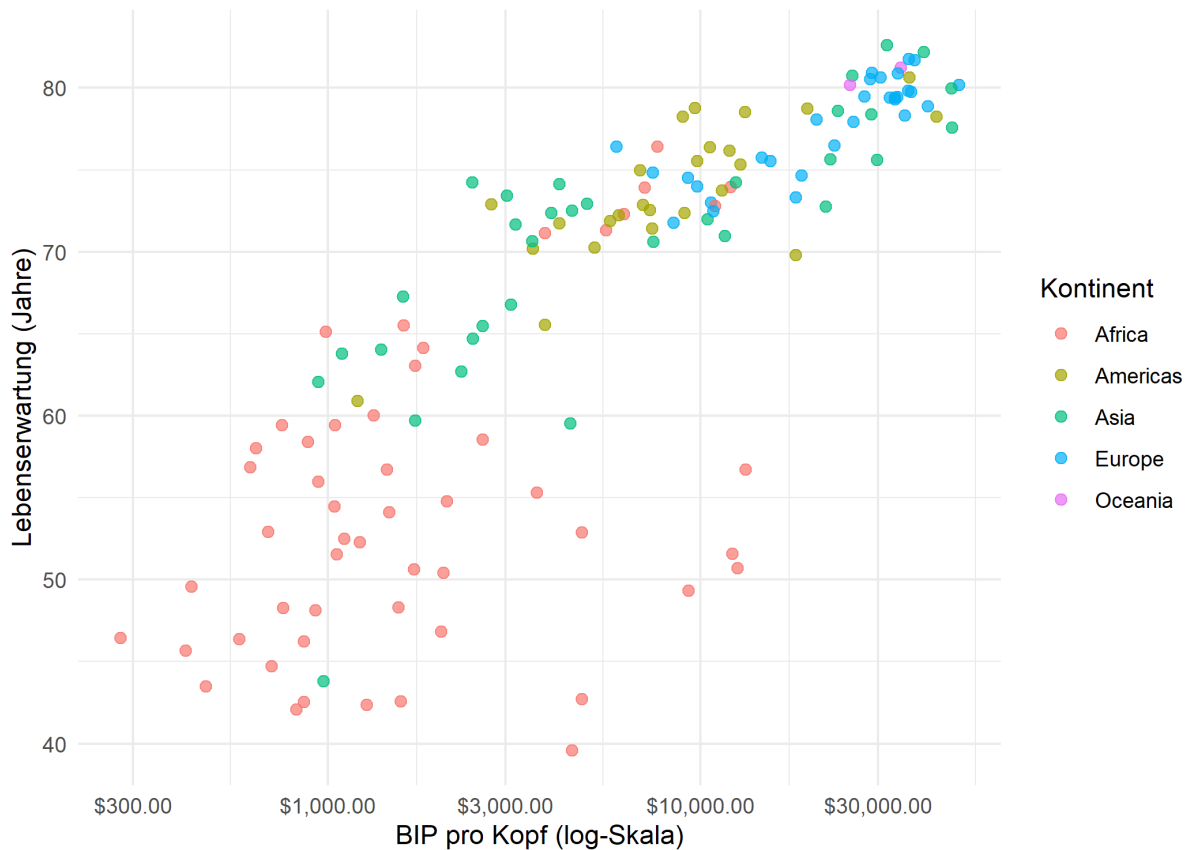
Man beginnt mit einem Streudiagramm von BIP pro Kopf gegen Lebenserwartung fuer das Jahr 2007, eingefaebrt nach Kontinent. Da BIP-Werte stark rechtsschief verteilt sind, verwendet man eine logarithmische x-Achse, um die Punkte gleichmaessiger zu verteilen.

```
gap07 <- gapminder::gapminder %>%
  filter(year == 2007)

p_base <- ggplot(gap07, aes(x = gdpPercap, y = lifeExp, color = continent)) +
  geom_point(alpha = 0.7, size = 2) +
  scale_x_log10(labels = scales::label_dollar()) +
  labs(
    x = "BIP pro Kopf (log-Skala)",
    y = "Lebenserwartung (Jahre)",
    color = "Kontinent"
  ) +
```

```
theme_minimal()
```

```
p_base
```

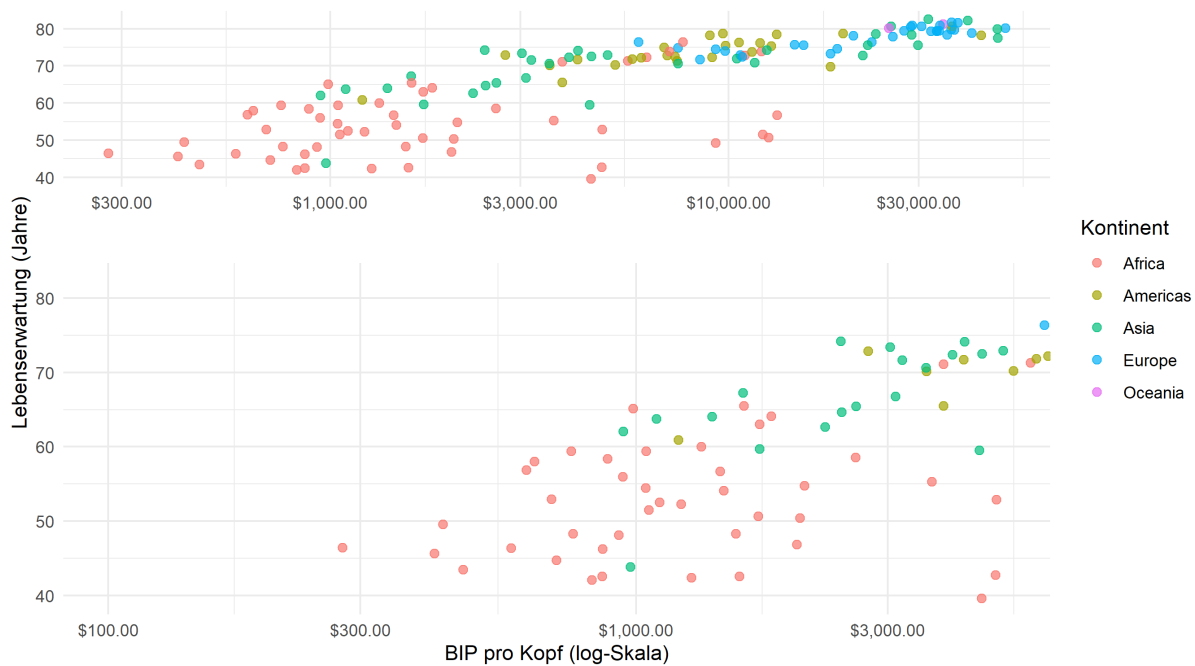


Selbst mit logarithmischer Skalierung ist der Bereich niedriger BIP-Werte (unter \$5.000) dicht gedrängt und schwer lesbar. Genau in dieser Situation zeigt `facet_zoom()` seine Staerke.

Einfacher Zoom

Die einfachste Verwendung gibt ein `xlim` an, um den interessierenden Bereich zu definieren. Das Ergebnis ist ein Zwei-Panel-Plot: das Original auf einer Seite und die vergrößerte Ansicht auf der anderen.

```
p_base +  
  facet_zoom(xlim = c(100, 5000))
```

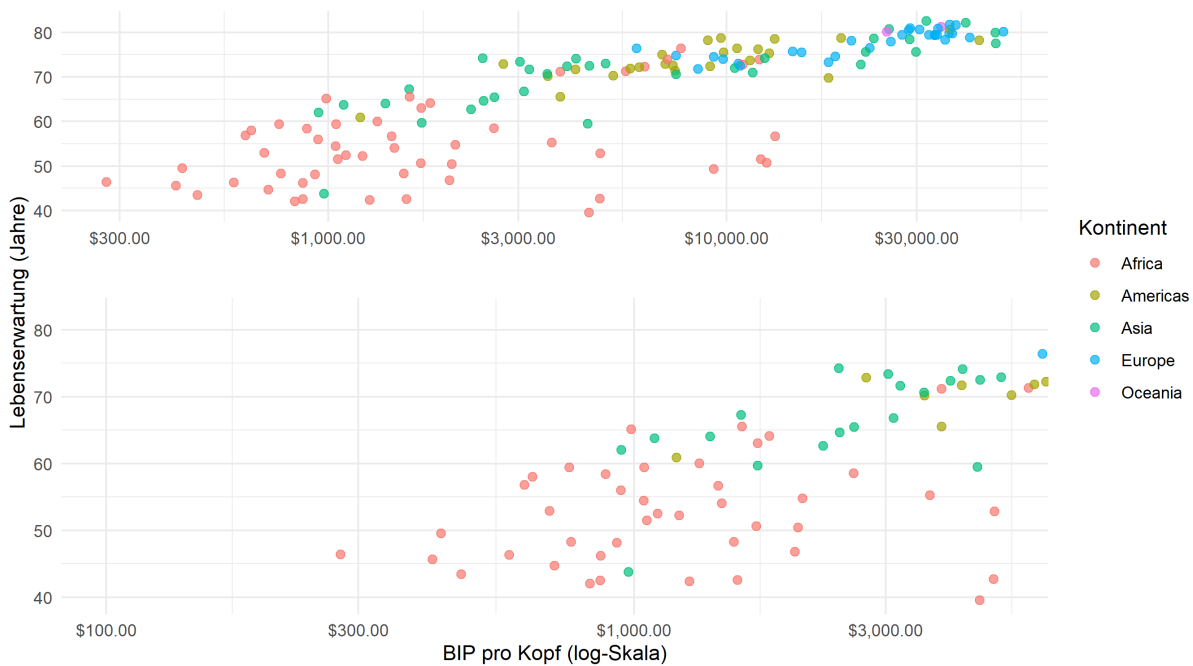


Das Zoom-Panel zeigt nur Laender mit einem BIP pro Kopf zwischen \$100 und \$5.000, was es deutlich einfacher macht, einzelne Punkte in diesem dichten Bereich zu unterscheiden. Ein schattiertes Rechteck im Uebersichtspanel zeigt an, welcher Bereich vergroessert wird.

Gestyler Zoom

Zwei nuetzliche Parameter steuern das Erscheinungsbild: `zoom.size` passt die relative Groesse des Zoom-Panels an (Werte groesser als 1 machen es groesser als die Uebersicht), und `show.area = TRUE` hebt den gezoomten Bereich im Uebersichtspanel durch ein schattiertes Rechteck hervor.

```
p_base +
  facet_zoom(
    xlim = c(100, 5000),
    zoom.size = 1.5,
    show.area = TRUE
  )
```



`facet_zoom()` akzeptiert auch einen `ylim`-Parameter fuer vertikales Zoomen, und beide koennen kombiniert werden, um in einen rechteckigen Teilbereich zu zoomen. Fuer die meisten Anwendungsfaelle reicht das Zoomen entlang einer einzelnen Achse aus und ist leichter zu interpretieren.

geom_mark_ellipse

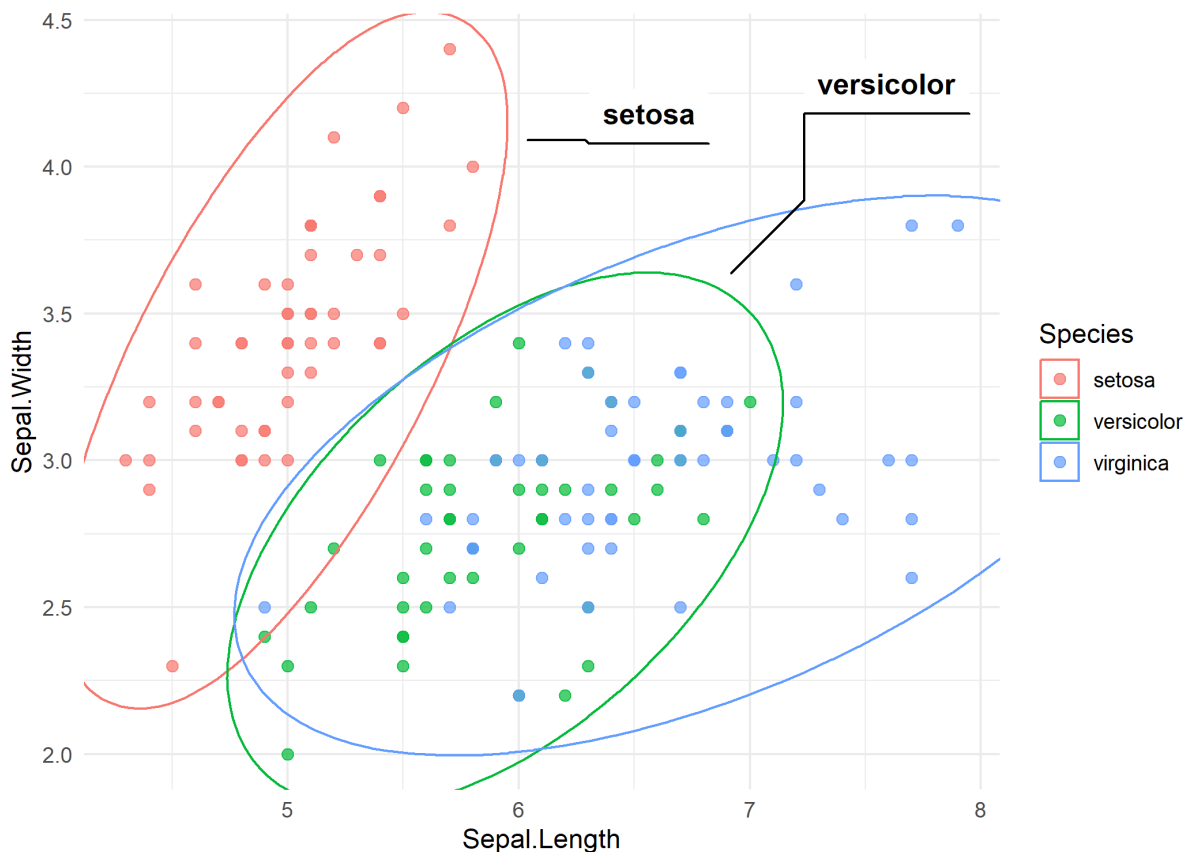
Wenn ein Streudiagramm deutliche Cluster enthaelt, kann es hilfreich sein, jede Gruppe visuell zu umranden. `geom_mark_ellipse()` zeichnet eine statistische Ellipse (basierend auf der Kovarianzmatrix) um die Punkte jeder Gruppe. Dies ist besonders effektiv fuer Praesentationen, bei denen man die Gruppentrennung hervorheben moechte.

Einfache Ellipse

Der iris-Datensatz enthaelt Messungen von Kelch- und Bluetenblattdimensionen fuer drei Arten. Man mappt `Species` sowohl auf `color` als auch auf `group` und fuegt Labels ueber die `label`-Aesthetik hinzu.

```
p_iris <- ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width, color = Species)) +
  geom_point(size = 2, alpha = 0.7) +
  theme_minimal()

p_iris +
  geom_mark_ellipse(aes(label = Species, group = Species))
```



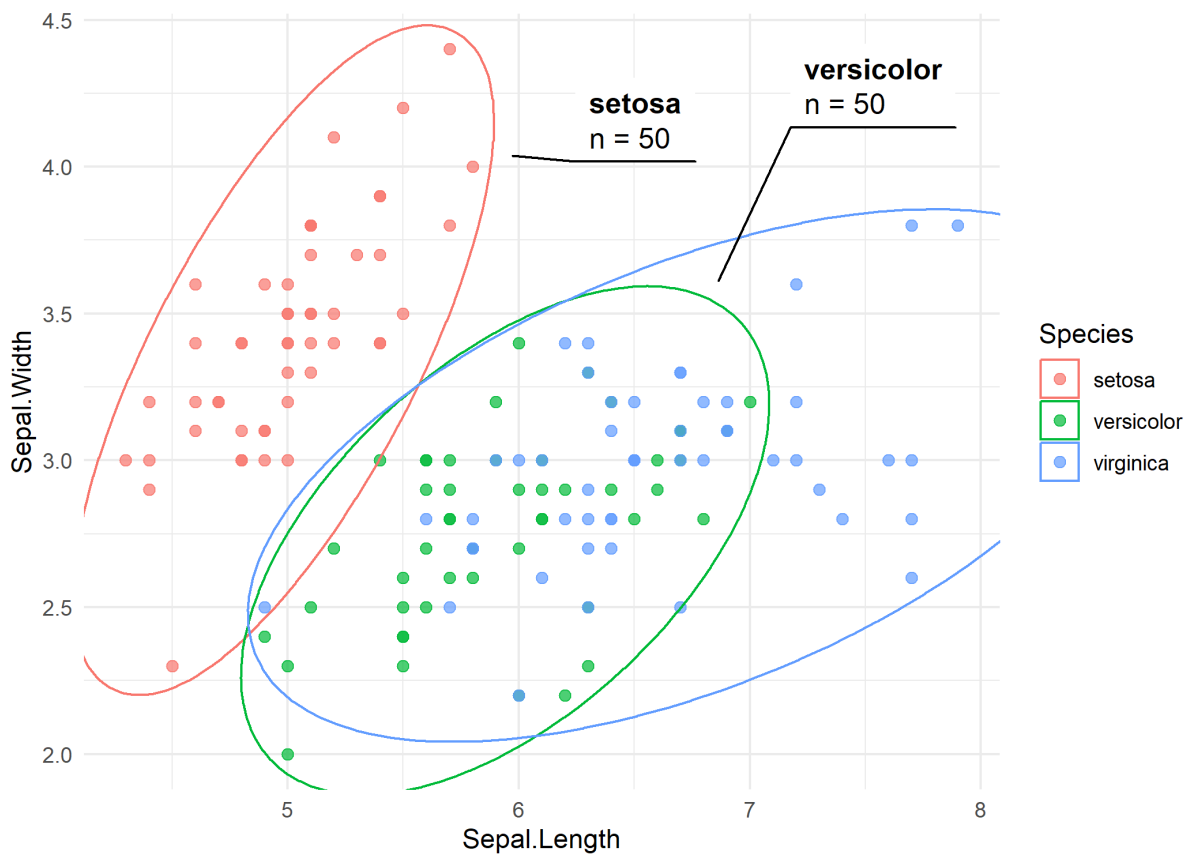
Jede Ellipse umschließt die Punkte einer Art, und ein Label wird ausserhalb der Ellipse mit einer Verbindungslinie platziert. Die Ellipsenform spiegelt die bivariate Verteilung jeder Gruppe wider - laengliche Ellipsen deuten auf korrelierte Variablen innerhalb dieser Gruppe hin.

Gestylte Ellipse

Mehrere Parameter steuern das Erscheinungsbild. `expand` fuegt Abstand um die Ellipse hinzu, `con.type` aendert den Verbindungslinienstil ("elbow", "straight" oder "none"), und die `description`-Aesthetik fuegt ein sekundaeres Label unterhalb des Hauptlabels hinzu. Hier ergaenzt man die Anzahl der Beobachtungen pro Art als Beschreibung.

```
iris_desc <- iris %>%
  group_by(Species) %>%
  mutate(desc = glue::glue("n = {n()}")) %>%
  ungroup()

ggplot(iris_desc, aes(x = Sepal.Length, y = Sepal.Width, color = Species)) +
  geom_point(size = 2, alpha = 0.7) +
  geom_mark_ellipse(
    aes(label = Species, group = Species, description = desc),
    expand = unit(3, "mm"),
    con.type = "straight"
  ) +
  theme_minimal()
```



Die geraden Verbindungslinien wirken aufgeräumter, und die Beschreibungszeile liefert zusätzlichen Kontext, ohne den Plot zu überladen.

geom_mark_hull

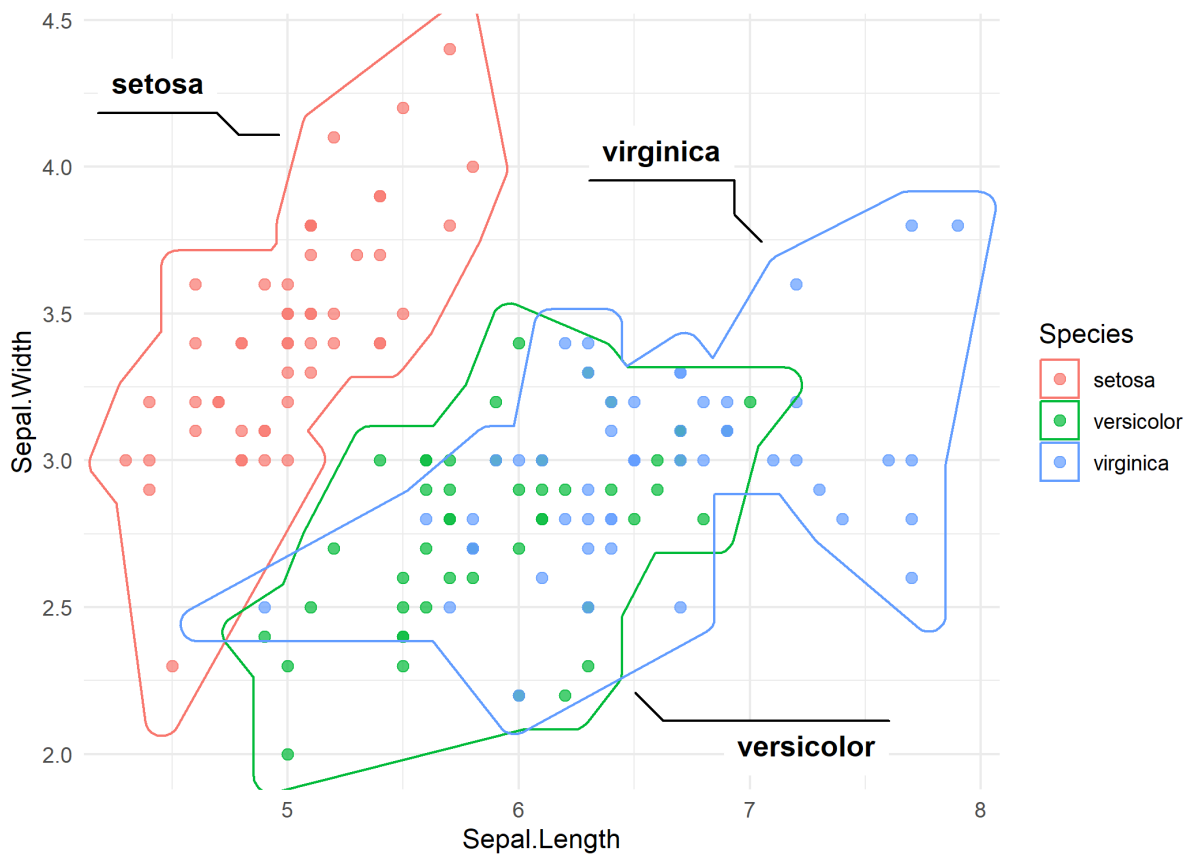
Während Ellipsen eine annähernd normale Verteilung voraussetzen, zeichnen konvexe Hüllen die tatsächliche Grenze der Punkte jeder Gruppe nach. Das macht

`geom_mark_hull()` zur besseren Wahl, wenn Gruppen unregelmässige Formen haben oder wenn man die präzise Ausdehnung der Daten hervorheben möchte, statt einer idealisierten statistischen Form.

Einfache Huelle

Die Syntax entspricht der von `geom_mark_ellipse()`:

```
p_iris +
  geom_mark_hull(aes(label = Species, group = Species))
```

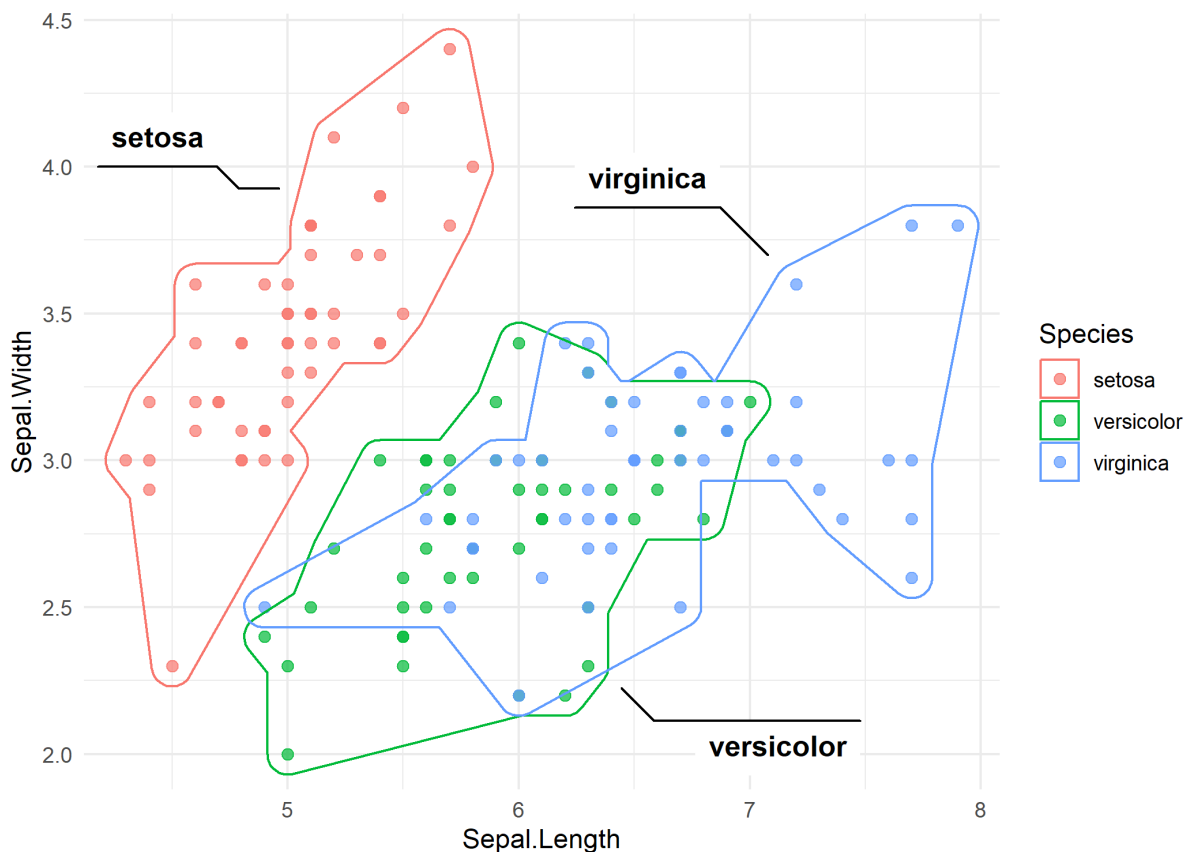


Die Huelle umschliesst eng die aeussersten Punkte jeder Gruppe. Im Vergleich zu den Ellipsen folgt die Huelle der tatsaechlichen Datengrenze - man beachte, wie die Setosa-Huelle kompakter ist, weil die Punkte eng beieinander liegen.

Gestylte Huelle

Die Parameter `expand` und `radius` steuern den Abstand und die Eckenrundung. Erhoehen beider Werte erzeugt eine glattere, optisch ansprechendere Form.

```
p_iris +
  geom_mark_hull(
    aes(label = Species, group = Species),
    expand = unit(3, "mm"),
    radius = unit(3, "mm")
  )
```



Die abgerundeten Ecken lassen die Huelen weniger kantig und polierter wirken, was fuer Praesentationen und Publikationen oft bevorzugt wird.

💡 Ellipse vs. Huelle vs. Rechteck - wann nutzt man was?

Das ggforce-Paket bietet drei Geoms zur Gruppenmarkierung, die jeweils fuer unterschiedliche Situationen geeignet sind:

- `geom_mark_ellipse()` : Am besten geeignet, wenn Gruppen annaehernd normalverteilt sind. Die Ellipse spiegelt die statistische Streuung und Orientierung jedes Clusters wider. Nuetzlich, um Verteilungseigenschaften zu vermitteln.
- `geom_mark_hull()` : Am besten geeignet, wenn man die tatsaechliche Ausdehnung der Daten hervorheben moechte, ohne Verteilungsannahmen zu treffen. Die Huelle folgt exakt den aeussersten Punkten (mit optionaler Glaettung).
- `geom_mark_rect()` : Die einfachste Option - ein rechteckiger Begrenzungsrahmen um jede Gruppe. Nuetzlich, wenn Gruppen gut getrennt sind und eine einfache Hervorhebung genuegt.

Alle drei akzeptieren dieselben Kern-Aesthetiken (`label` , `group` , `description`) und Styling-Parameter (`expand` , `con.type`), sodass man leicht zwischen ihnen wechseln kann.

Bibliography