

Package ‘gghotelling’

August 9, 2026

Title Hotelling's T² Ellipses and Outlier Detection for 'ggplot2'

Version 0.2.1

Description Calculate Hotelling's T² ellipses and detect multivariate outliers both for base R plots and 'ggplot2' plots. Optionally, uses robust covariance estimation to reduce the influence of outliers on the ellipses. Also included: bagplots, kernel density plots and outlier diagnostic plots.

URL <https://github.com/january3/gghotelling/>,
<https://january3.github.io/gghotelling/>

License MIT + file LICENSE

Encoding UTF-8

Imports ggplot2, grDevices, MASS, mrfDepth, robustbase

Suggests testthat (>= 3.0.0), ggrepel, knitr, rmarkdown, HDclassif, spelling, cowplot

Config/testthat/edition 3

VignetteBuilder knitr

Language en-US

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author January Weiner [aut, cre, cph] (ORCID:
<<https://orcid.org/0000-0003-1438-7819>>)

Maintainer January Weiner <january.weiner@gmail.com>

Repository CRAN

Date/Publication 2026-08-09 06:50:13 UTC

Contents

autoplot.prcomp	2
GeomHull	3
GeomKDE	5

hotelling_ellipse	7
outliers	8
StatBag	10
StatKDE	12
StatOutliers	14
Index	17

autoplot.prcomp	<i>Automatic plotting of PCA objects</i>
-----------------	--

Description

Automatic plotting of PCA objects

Usage

```
## S3 method for class 'prcomp'
autoplot(object, dims = c(1, 2), biplot = FALSE, group = NULL, ...)

## S3 method for class 'prcomp'
autolayer(
  object,
  dims = c(1, 2),
  group = NULL,
  labels = NULL,
  type = c("t2data", "t2mean", "c2data"),
  outliers = TRUE,
  level = 0.95,
  ...
)
```

Arguments

object	An object of prcomp class
dims	Dimensions to plot
biplot	whether to show the loadings as well as the scores
group	Groups of the data to be shown on the plot
...	ignored
labels	optionally, a vector of labels for showing the outliers. If NULL, the outliers will be identified by row number.
type	The type of the coverage / confidence area shown by autolayer.prcomp(), can be one of t2data (T2 Hotelling coverage), c2data (chi-squared coverage) or t2mean (T-squared based confidence area for the group mean).
outliers	if TRUE, label the outliers.
level	Either coverage probability (for type = "t2data" and "c2data") or confidence level (for type = "t2mean").

Details

The functions `autoplot.prcomp` and `autolayer.prcomp` are for automatic plotting of `prcomp` objects, similar to `ggfortify::autoplot.prcomp`.

Note, however, that the `group` parameter is a vector of the same length as the number of rows in the PCA object, rather than a column name in the data frame.

Value

A `ggplot2` object

Examples

```
pca <- prcomp(iris[,1:4], scale.=TRUE)

library(ggplot2)
autoplot(pca, group = iris$Species) +
  autolayer(pca, group = iris$Species)

# show the 90% confidence area for the group means
autoplot(pca, group = iris$Species) +
  autolayer(pca, group = iris$Species,
            type="t2mean", level = 0.90,
            outliers = FALSE)
```

GeomHull

Draw a convex hull around a group of points

Description

Draw a convex hull around a group of points

Usage

```
geom_hull(
  mapping = NULL,
  data = NULL,
  position = "identity",
  ...,
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE
)
```

Arguments

mapping	Set of aesthetic mappings created by aes() . If specified and <code>inherit.aes = TRUE</code> (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping.
data	The data to be displayed in this layer. There are three options: If <code>NULL</code>, the default, the data is inherited from the plot data as specified in the call to ggplot(). A <code>data.frame</code>, or other object, will override the plot data. All objects will be fortified to produce a data frame. See fortify() for which variables will be created. A function will be called with a single argument, the plot data. The return value must be a <code>data.frame</code>, and will be used as the layer data. A function can be created from a formula (e.g. <code>~ head(.x, 10)</code>).
position	A position adjustment to use on the data for this layer. This can be used in various ways, including to prevent overplotting and improving the display. The position argument accepts the following: • The result of calling a position function, such as position_jitter(). This method allows for passing extra arguments to the position. • A string naming the position adjustment. To give the position as a string, strip the function name of the <code>position_</code> prefix. For example, to use position_jitter(), give the position as "jitter". • For more information and other ways to specify the position, see the layer position documentation.
...	Additional parameters passed to underlying <code>ggplot2::geom_polygon()</code> or to <code>ggplot2::layer()</code> .
na.rm	Logical. Should missing values be removed? Default is <code>FALSE</code> .
show.legend	logical. Should this layer be included in the legends? <code>NA</code> , the default, includes if any aesthetics are mapped. <code>FALSE</code> never includes, and <code>TRUE</code> always includes. It can also be a named logical vector to finely select the aesthetics to display. To include legend keys for all levels, even when no data exists, use <code>TRUE</code> . If <code>NA</code> , all levels are shown in legend, but unobserved levels are omitted.
inherit.aes	If <code>FALSE</code> , overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. annotation_borders() .

Value

A `ggplot2` geom layer that can be added to a `ggplot` object.

Examples

```
# nothing
library(ggplot2)
ggplot(iris, aes(Sepal.Length, Sepal.Width, color=Species)) +
  geom_hull(mapping = aes(fill = Species), alpha=.1) +
  geom_point()
```

Description

Hotelling ellipses for ggplot2

Usage

```
geom_hotelling(  
  mapping = NULL,  
  data = NULL,  
  position = "identity",  
  ...,  
  level = 0.95,  
  type = "t2data",  
  robust = FALSE,  
  npoints = 100,  
  na.rm = FALSE,  
  show.legend = NA,  
  inherit.aes = TRUE  
)
```

Arguments

- | | |
|----------|---|
| mapping | Set of aesthetic mappings created by aes() . If specified and <code>inherit.aes = TRUE</code> (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping. |
| data | <p>The data to be displayed in this layer. There are three options:</p> <p>If <code>NULL</code>, the default, the data is inherited from the plot data as specified in the call to ggplot().</p> <p>A <code>data.frame</code>, or other object, will override the plot data. All objects will be fortified to produce a data frame. See fortify() for which variables will be created.</p> <p>A function will be called with a single argument, the plot data. The return value must be a <code>data.frame</code>, and will be used as the layer data. A function can be created from a formula (e.g. <code>~ head(.x, 10)</code>).</p> |
| position | <p>A position adjustment to use on the data for this layer. This can be used in various ways, including to prevent overplotting and improving the display. The position argument accepts the following:</p> <ul style="list-style-type: none"> • The result of calling a position function, such as <code>position_jitter()</code>. This method allows for passing extra arguments to the position. • A string naming the position adjustment. To give the position as a string, strip the function name of the <code>position_</code> prefix. For example, to use <code>position_jitter()</code>, give the position as "jitter". |

	For more information and other ways to specify the position, see the layer position documentation.
...	Additional parameters passed to underlying <code>ggplot2::geom_polygon()</code> or to <code>ggplot2::layer()</code> .
level	Either coverage probability (for type = "t2data" or "c2data") or confidence level (for type = "t2mean").
type	t2data - Hotelling T2 data ellipse; t2mean - Hotelling confidence interval for the mean; c2data - normal data ellipse (using chi squared distribution).
robust	If TRUE, then robust estimates of mean and covariance are used
npoints	Number of points to estimate
na.rm	Logical. Should missing values be removed? Default is FALSE.
show.legend	logical. Should this layer be included in the legends? NA, the default, includes if any aesthetics are mapped. FALSE never includes, and TRUE always includes. It can also be a named logical vector to finely select the aesthetics to display. To include legend keys for all levels, even when no data exists, use TRUE. If NA, all levels are shown in legend, but unobserved levels are omitted.
inherit.aes	If FALSE, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. annotation_borders() .

Details

This geom adds data or confidence ellipses to the plot. See `hotelling_ellipse()` documentation for more information.

Value

A `ggplot2` geom layer that can be added to a `ggplot` object.

Examples

```
pca <- prcomp(iris[, 1:4], scale.=TRUE)
df <- cbind(iris, pca$x)
library(ggplot2)

ggplot(df, aes(PC1, PC2)) +
  geom_hotelling(level=.99) +
  geom_point()

ggplot(df, aes(PC1, PC2, color=Species)) +
  geom_hotelling() +
  geom_point()

ggplot(df, aes(PC1, PC2, color=Species)) +
  geom_hotelling(alpha=0.1, aes(fill = Species)) +
  geom_point()

# compare the robust and regular approaches:
```

```
ggplot(df, aes(PC1, PC2, color=Species)) +
  geom_hotelling() +
  geom_hotelling(robust = TRUE, linetype = "dashed") +
  geom_point()
```

hotelling_ellipse	<i>Calculate Hotelling or data ellipse around some points</i>
-------------------	---

Description

Calculate Hotelling or data ellipse around some points

Usage

```
hotelling_ellipse(
  x,
  level = 0.95,
  npoints = 100,
  type = c("t2data", "t2mean", "c2data"),
  robust = FALSE
)
```

Arguments

x	A two-column matrix or data frame like object
level	Either coverage probability (for type = "t2data" or "c2data") or confidence level (for type = "t2mean").
npoints	Number of points to estimate
type	t2data - Hotelling T2 data ellipse; t2mean - Hotelling confidence interval for the mean; c2data - normal data ellipse (using chi squared distribution).
robust	If TRUE, then robust estimates of mean and covariance are used

Details

Calculate the T2 Hotelling ellipse or a data ellipse for the given coverage probability. There are three types of ellipses which can be plotted:

- T2 Hotelling data ellipses, showing data coverage (like the type="t" version of stat_ellipse)
- normal multivariate distribution ellipses (like the type="norm" version of the stat_ellipse) which use Mahalanobis distance and chi-squared statistic
- T2 Hotelling confidence ellipses of the group means.

The latter (for group means) correspond to the confidence interval for the mean in the univariate world, so the ellipses are very small (depending on the number of points).

The function can also use a robust estimator of location and scatter using the `covMcd` function, which uses the Maximum Covariance Determinant (MCD) estimator. Note that while this results in ellipses which are more resistant to outliers, the interpretation slightly changes, as the T2 statistic used is only an approximation in this case. In other words, use it for visualisation and QC, but not for statistical testing.

Value

A two-column matrix or data frame with `npoints` rows, containing the coordinates of the ellipse. The columns are named "x" and "y".

See Also

`outliers()` for calculating per-point based statistic and `geom_hotelling()` for plotting of the ellipse with `ggplot`

Examples

```
df <- iris[ iris$Species == "setosa", 1:2 ]
eli <- hotelling_ellipse(df)
plot(df[,1], df[,2])
lines(eli)
```

outliers

Calculate the T2/D2 statistic for individual points

Description

Calculate the T2/D2 statistic or Mahalanobis distance for individual points. For each point, it calculates the squared Mahalanobis distance from the center of the data, and compares it to the critical T2 or chi-squared value for the given coverage probability.

Usage

```
outliers(
  x,
  level = 0.95,
  robust = FALSE,
  type = c("t2data", "t2mean", "c2data")
)

plot_outliers(
  x,
  level = 0.95,
  robust = FALSE,
  type = c("t2data", "t2mean", "c2data"),
```



```

    labels = NULL
  )

```

Arguments

<code>x</code>	A matrix or data frame with two columns
<code>level</code>	Either coverage probability (for <code>type = "t2data"</code> or <code>"c2data"</code>) or confidence level (for <code>type = "t2mean"</code>).
<code>robust</code>	If TRUE, then robust estimates of mean and covariance are used
<code>type</code>	what type of statistic should be calculated; can be <code>t2data</code> (for data coverage), <code>t2mean</code> (for difference from a mean) or <code>"c2data"</code> (for coverage calculated with the chi squared statistic)
<code>labels</code>	Optional labels to use on the plot instead of rownames

Details

The function can use a robust estimator of location and scatter using the [covMcd](#) function, which uses the Maximum Covariance Determinant (MCD) estimator. Note that while this results in ellipses which are more resistant to outliers, the interpretation slightly changes, as the T2 statistic used is only an approximation in this case. In other words, use it for visualisation and QC, but not for statistical testing.

Value

A data frame with one row per point including the columns `d2` (squared mahalanobis distance or robust squared distance), `t2crit` (critical T squared value for the given level), `c2crit` (critical X squared value for the given level) and `is_outlier` (logical, whether `d2 > t2crit` or `d2 > c2crit`, depending on type).

See Also

[hotelling_ellipse](#) for more information on the differences between `t2data`, `t2mean` and `c2data` modes.

Examples

```

library(ggplot2)

pca <- prcomp(iris[, 1:4], scale.=TRUE)
pca_df <- cbind(iris, pca$x)
outlier_stats <- outliers(pca_df[, c("PC1", "PC2")], level = 0.95)

# use plot_outliers() to directly plot them
plot_outliers(pca_df[, c("PC1", "PC2")], level = 0.95)

```

StatBag

*Bag plots***Description**

Bag plots for ggplot2

Usage

```
geom_bag(
  mapping = NULL,
  data = NULL,
  position = "identity",
  ...,
  what = c("bag", "loop"),
  type = c("hdepth", "projdepth", "sprojdepth", "dprojdepth"),
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE
)
```

Arguments

- | | |
|----------|--|
| mapping | Set of aesthetic mappings created by aes() . If specified and <code>inherit.aes = TRUE</code> (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping. |
| data | <p>The data to be displayed in this layer. There are three options:</p> <p>If <code>NULL</code>, the default, the data is inherited from the plot data as specified in the call to ggplot().</p> <p>A <code>data.frame</code>, or other object, will override the plot data. All objects will be fortified to produce a data frame. See fortify() for which variables will be created.</p> <p>A function will be called with a single argument, the plot data. The return value must be a <code>data.frame</code>, and will be used as the layer data. A function can be created from a formula (e.g. <code>~ head(.x, 10)</code>).</p> |
| position | <p>A position adjustment to use on the data for this layer. This can be used in various ways, including to prevent overplotting and improving the display. The <code>position</code> argument accepts the following:</p> <ul style="list-style-type: none"> • The result of calling a position function, such as position_jitter(). This method allows for passing extra arguments to the position. • A string naming the position adjustment. To give the position as a string, strip the function name of the <code>position_</code> prefix. For example, to use position_jitter(), give the position as "jitter". • For more information and other ways to specify the position, see the layer position documentation. |

<code>...</code>	Additional parameters passed to underlying <code>ggplot2::geom_polygon()</code> or to <code>ggplot2::layer()</code> .
<code>what</code>	What to plot. Either "bag" (the inner bag region) or "loop" (the outer loop).
<code>type</code>	Determines the depth function used to construct the bagplot: "hdepth" for half-space depth, "projdepth" for projection depth, "sprojdepth" for skewness-adjusted projection depth and "dprojdepth" for directional projection depth. Defaults to "hdepth".
<code>na.rm</code>	Logical. Should missing values be removed? Default is FALSE.
<code>show.legend</code>	logical. Should this layer be included in the legends? NA, the default, includes if any aesthetics are mapped. FALSE never includes, and TRUE always includes. It can also be a named logical vector to finely select the aesthetics to display. To include legend keys for all levels, even when no data exists, use TRUE. If NA, all levels are shown in legend, but unobserved levels are omitted.
<code>inherit.aes</code>	If FALSE, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. annotation_borders() .

Details

Bag plots are 2-dimensional generalizations of box plots. This geom adds a bag region (which contains 50% of the data points, just like the "box" in "box plots") to the plot, plus the "loop" which can be used to identify potential outliers.

`geom_bag()` is just a wrapper around the [mrfDepth::compBagplot\(\)](#) function which actually does the calculations, and the arguments are passed on to that function.

The geom either plots the inner "bag" or the outer "loop", but not both. If you need both, you can add two `geom_bag` layers to your `ggplot` object.

Value

A `ggplot2` geom layer that can be added to a `ggplot` object.

Examples

```
library(ggplot2)
ggplot(iris, aes(x = Sepal.Width, y = Sepal.Length, color = Species)) +
  geom_point() +
  geom_bag(aes(fill = Species), alpha = .1, what = "loop") +
  geom_bag(aes(fill = Species), alpha = .3)
```

Description

Kernel density estimate contours for ggplot2

Usage

```
geom_kde(
  mapping = NULL,
  data = NULL,
  position = "identity",
  ...,
  coverage = 0.95,
  n = 100,
  show.legend = NA,
  inherit.aes = TRUE
)
```

Arguments

- | | |
|----------|---|
| mapping | Set of aesthetic mappings created by aes() . If specified and <code>inherit.aes = TRUE</code> (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping. |
| data | <p>The data to be displayed in this layer. There are three options:</p> <p>If <code>NULL</code>, the default, the data is inherited from the plot data as specified in the call to ggplot().</p> <p>A <code>data.frame</code>, or other object, will override the plot data. All objects will be fortified to produce a data frame. See fortify() for which variables will be created.</p> <p>A function will be called with a single argument, the plot data. The return value must be a <code>data.frame</code>, and will be used as the layer data. A function can be created from a formula (e.g. <code>~ head(.x, 10)</code>).</p> |
| position | <p>A position adjustment to use on the data for this layer. This can be used in various ways, including to prevent overplotting and improving the display. The position argument accepts the following:</p> <ul style="list-style-type: none"> • The result of calling a position function, such as position_jitter(). This method allows for passing extra arguments to the position. • A string naming the position adjustment. To give the position as a string, strip the function name of the <code>position_</code> prefix. For example, to use position_jitter(), give the position as "jitter". • For more information and other ways to specify the position, see the layer position documentation. |

...	Additional parameters passed to underlying <code>ggplot2::geom_polygon()</code> or to <code>ggplot2::layer()</code> .
coverage	What fraction of the points should be enclosed by the contour
n	Number of grid points in each direction (passed to <code>kde2d</code>)
show.legend	logical. Should this layer be included in the legends? NA, the default, includes if any aesthetics are mapped. FALSE never includes, and TRUE always includes. It can also be a named logical vector to finely select the aesthetics to display. To include legend keys for all levels, even when no data exists, use TRUE. If NA, all levels are shown in legend, but unobserved levels are omitted.
inherit.aes	If FALSE, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. <code>annotation_borders()</code> .

Details

This function uses the `kde2d` function to estimate density along a grid, and then plots a contour for a selected coverage.

Value

A ggplot2 geom layer that can be added to a ggplot object.

See Also

`geom_hull()` for simple convex contours and `geom_hotelling()` for Hotelling T2 ellipses.

Examples

```
library(ggplot2)

# following shows a comparison
# between a convex hull and KDE contour
df <- data.frame(x=rnorm(500) + 5)
df$y <- df$x^5 + rnorm(500)*500
ggplot(df, aes(x=x, y=y)) +
  geom_point()+
  geom_hull(color = "grey") +
  geom_kde(color="red", linewidth=1)

# we can also achieve similar effects to geom_contour
pca <- prcomp(iris[, 1:4], scale.=TRUE)
df <- cbind(iris, pca$x)

# following is similar to geom_contour
pca <- prcomp(iris[, 1:4], scale.=TRUE)
df <- cbind(iris, pca$x)
p <- ggplot(df, aes(x = PC1, y = PC2, color=Species))
for(i in seq(.05, .95, length.out = 10)) {
  p <- p + geom_kde(aes(fill = Species), alpha = .05, coverage = i, n = 100)
}
```

```
p + geom_point()
```

StatOutliers

Calculate per-point Hotelling statistic

Description

Calculate per-point Hotelling statistic for use in ggplot

Usage

```
stat_outliers(
  mapping = NULL,
  data = NULL,
  geom = "point",
  position = "identity",
  ...,
  type = c("t2data", "c2data"),
  level = 0.95,
  outlier_only = FALSE,
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE
)
```

Arguments

- | | |
|---------|---|
| mapping | Set of aesthetic mappings created by aes() . If specified and <code>inherit.aes = TRUE</code> (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping. |
| data | <p>The data to be displayed in this layer. There are three options:</p> <p>If <code>NULL</code>, the default, the data is inherited from the plot data as specified in the call to ggplot().</p> <p>A <code>data.frame</code>, or other object, will override the plot data. All objects will be fortified to produce a data frame. See fortify() for which variables will be created.</p> <p>A function will be called with a single argument, the plot data. The return value must be a <code>data.frame</code>, and will be used as the layer data. A function can be created from a formula (e.g. <code>~ head(.x, 10)</code>).</p> |
| geom | <p>The geometric object to use to display the data for this layer. When using a <code>stat_*()</code> function to construct a layer, the <code>geom</code> argument can be used to override the default coupling between stats and geoms. The <code>geom</code> argument accepts the following:</p> <ul style="list-style-type: none"> • A <code>Geom</code> ggproto subclass, for example <code>GeomPoint</code>. |

	• A string naming the geom. To give the geom as a string, strip the function name of the <code>geom_</code> prefix. For example, to use <code>geom_point()</code>, give the geom as "point". • For more information and other ways to specify the geom, see the layer geom documentation.
position	A position adjustment to use on the data for this layer. This can be used in various ways, including to prevent overplotting and improving the display. The position argument accepts the following: • The result of calling a position function, such as <code>position_jitter()</code>. This method allows for passing extra arguments to the position. • A string naming the position adjustment. To give the position as a string, strip the function name of the <code>position_</code> prefix. For example, to use <code>position_jitter()</code>, give the position as "jitter". • For more information and other ways to specify the position, see the layer position documentation.
...	Additional parameters passed to underlying <code>ggplot2::geom_polygon()</code> or to <code>ggplot2::layer()</code> .
type	t2data - Hotelling T2 data ellipse; t2mean - Hotelling confidence interval for the mean; c2data - normal data ellipse (using chi squared distribution).
level	Either coverage probability (for type = "t2data" or "c2data") or confidence level (for type = "t2mean").
outlier_only	Only return the statistic for outliers
na.rm	Logical. Should missing values be removed? Default is FALSE.
show.legend	logical. Should this layer be included in the legends? NA, the default, includes if any aesthetics are mapped. FALSE never includes, and TRUE always includes. It can also be a named logical vector to finely select the aesthetics to display. To include legend keys for all levels, even when no data exists, use TRUE. If NA, all levels are shown in legend, but unobserved levels are omitted.
inherit.aes	If FALSE, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. annotation_borders() .

Details

This calculates the Hotelling statistic for each point in the plot, group-wise. This allows to use the statistics `is_outlier`, `d2` (the squared Mahalanobis distance), `c2crit` (critical Chi-squared value for the specified level) and `t2crit` (critical Hotelling T2 value for the squared Mahalanobis distance) to be used as graphical parameters, e.g. for coloring the points (see Examples below) using the `ggplot2::after_stat()` function.

The `is_outlier` is simply either `d2 > t2crit` (if type="t2data" or type="t2mean") or `d2 > c2crit` (if type="c2data").

The `type` argument chooses between the regular Hotelling statistic (with type="t2data") or the Chi-squared statistic (with type="c2data"). The Hotelling statistic for the mean is not allowed as it makes no sense in the context of the outliers.

The function is a wrapper around the `outliers()` function, which does the actual calculations, and the parameters passed (`level`, `type`, `robust`) are passed on to that function. For example, to calculate robust statistic, use `stat_outliers(robust=TRUE)`.

Value

A stat layer that can be added to a ggplot object.

Examples

```
pca <- prcomp(iris[, 1:4], scale.=TRUE)
df <- cbind(iris, pca$x)

library(ggplot2)
ggplot(df, aes(PC1, PC2, group=Species)) +
  geom_hotelling(alpha=0.1, aes(fill = Species)) +
  scale_color_manual(values=c("TRUE"="red", "FALSE"="grey")) +
  stat_outliers(aes(color = after_stat(is_outlier)))

ggplot(df, aes(PC1, PC2, group=Species)) +
  geom_hotelling(alpha=0.1, level = .75, aes(fill = Species)) +
  stat_outliers(level = .75,
                size=2,
                aes(shape = Species,
                    color = after_stat(d2)))

# label the outliers
# note that you need to add the label aesthetics for the label geom to
# work
ggplot(df, aes(PC1, PC2, group=Species, label=rownames(df))) +
  geom_hotelling(level = 0.75, alpha=0.1, aes(fill = Species)) +
  geom_point(aes(color = Species)) +
  stat_outliers(level = .75, geom="label",
                outlier_only = TRUE)
```


Index

`aes()`, [4](#), [5](#), [10](#), [12](#), [14](#)
`annotation_borders()`, [4](#), [6](#), [11](#), [13](#), [15](#)
`autolayer.prcomp` (`autoplot.prcomp`), [2](#)
`autoplot.prcomp`, [2](#)

`covMcd`, [8](#), [9](#)

`fortify()`, [4](#), [5](#), [10](#), [12](#), [14](#)

`geom_bag` (`StatBag`), [10](#)
`geom_hotelling` (`GeomKDE`), [5](#)
`geom_hotelling()`, [8](#), [13](#)
`geom_hull` (`GeomHull`), [3](#)
`geom_hull()`, [13](#)
`geom_kde` (`StatKDE`), [12](#)
`GeomBag` (`StatBag`), [10](#)
`GeomHotelling` (`GeomKDE`), [5](#)
`GeomHull`, [3](#)
`GeomKDE`, [5](#)
`ggplot()`, [4](#), [5](#), [10](#), [12](#), [14](#)
`ggplot2::geom_polygon()`, [13](#)
`ggplot2::layer()`, [13](#)

`hotelling_ellipse`, [7](#), [9](#)

`kde2d`, [13](#)

`layer geom`, [15](#)
`layer position`, [4](#), [6](#), [10](#), [12](#), [15](#)

`mrfDepth::compBagplot()`, [11](#)

`outliers`, [8](#)
`outliers()`, [8](#)

`plot_outliers` (`outliers`), [8](#)

`stat_outliers` (`StatOutliers`), [14](#)
`StatBag`, [10](#)
`StatHotelling` (`GeomKDE`), [5](#)
`StatHull` (`GeomHull`), [3](#)
`StatKDE`, [12](#)
`StatOutliers`, [14](#)