```
R version 4.4.0 (2024-04-24) -- "Puppy Cup"
Copyright (C) 2024 The R Foundation for Statistical Computing
Platform: aarch64-apple-darwin20

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[R.app GUI 1.80 (8376) aarch64-apple-darwin20]

[History restored from /Users/alperkaragol/.Rapp.history]

> rm(list = ls())
> library(ggplot2)
> library(ggrepel)
>
> # Original data
> x <-
c(-1149.5,-680.6,-131.9,-1075,-760.6,-156.2,-226.7,-150,-144.7,-3455.6,-482,-515.6,-456.4,-1290.2,-168,-177.5,-1111,
-226.3,-1114.4,-324.3,-232.2,-164.4,-1089.4,-991)
> y <-
c(3919,2501,369,3717,2811,566,784,488,446,12006,1313,1351,1353,4608,629,607,3698,726,2420,695,592,424,2698,2471)
>
> # Labels for outliers
> labels <- c("EAA1 (Canonical)", "A0A7P0Z4R4", "A0A7P0T9Z4", "A0A087X0U3", "A0A7P0T8Q1", "E7EUV6", "E7EUS7",
"A0A7P0T9A4", "A0A7P0T807",
+              "EAA2 (Canonical)", "C9J9N5", "A0A2R8Y4D1", "A0A2R8Y642", "A0A2R8Y4N0", "A0A2R8YHI4", "H0YEB1",
+              "EAA3 (Canonical)", "H0Y7R2",
+              "EAA4 (Canonical)", "M0R106", "M0R2V7", "M0QY32",
+              "EAA5 (Canonical)", "F1T0D4")
>
> # Create a data frame
> df <- data.frame(x = x, y = y, label = labels)
>
> # Calculate Pearson correlation
> cor_result <- cor.test(x, y, alternative = "two.sided", method = "pearson", exact=FALSE)
> print(cor_result)

    Pearson's product-moment correlation

data:  x and y
t = -28.495, df = 22, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.9943337 -0.9690438
sample estimates:
       cor
-0.9867219

>
> # Function to identify pair outliers using Tukey's fence method
> identify_outliers <- function(x, y, k = 1.5) {
+   # Calculate Mahalanobis distances
+   center <- c(mean(x), mean(y))
+   cov_matrix <- cov(cbind(x, y))
+   mahalanobis_dist <- mahalanobis(cbind(x, y), center, cov_matrix)
+
+   # Calculate Tukey's fences
+   q1 <- quantile(mahalanobis_dist, 0.25)
+   q3 <- quantile(mahalanobis_dist, 0.75)
+   iqr <- q3 - q1
+   lower_fence <- q1 - k * iqr
+   upper_fence <- q3 + k * iqr
+
+   # Identify outliers
+   outliers <- mahalanobis_dist > upper_fence | mahalanobis_dist < lower_fence
+   return(outliers)
+ }
>
> # Identify outliers
> df$outlier <- identify_outliers(df$x, df$y)
>
> # Print number of outliers
> num_outliers <- sum(df$outlier)
> cat("Number of pair outliers detected:", num_outliers, "\n")
Number of pair outliers detected: 5
>
> # Fit linear regression model
> lm_model <- lm(y ~ x, data = df)
>
> # Create ggplot
> p <- ggplot(df, aes(x = x, y = y)) +
+   geom_point(aes(color = outlier, shape = outlier), size = 2.5) +  # Smaller data points
+   geom_smooth(method = "lm", se = FALSE, color = "darkgreen", linetype = "solid", size = 0.5) +  # Continuous line
+   scale_color_manual(values = c("steelblue", "red"),
+                      labels = c("Normal", "Outlier"),
+                      name = "Data Points") +
+   scale_shape_manual(values = c(16, 17),
+                      labels = c("Normal", "Outlier"),
+                      name = "Data Points") +
+   labs(title = " ",
+        x = " RNA structure MFE (kcal/mol)", y = " cDNA Length (nt)") +
+   theme_minimal(base_size = 13) +
+   theme(
+     legend.position = c(1, 1),  # Move legend to upper right corner
+     legend.justification = c(1, 1),   # Align legend to top-right
+     legend.box.just = "right",
+     legend.margin = margin(2, 2, 5, 5),
+     legend.background = element_rect(fill = "white", color = "black", size = 0.5),
+     panel.grid.major = element_line(color = "gray80", size = 0.2),
+     panel.grid.minor = element_line(color = "gray90", size = 0.05)
+   )
Warning messages:
1: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
ℹ Please use `linewidth` instead.
This warning is displayed once every 8 hours.
Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
2: The `size` argument of `element_rect()` is deprecated as of ggplot2 3.4.0.
ℹ Please use the `linewidth` argument instead.
This warning is displayed once every 8 hours.
Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
3: The `size` argument of `element_line()` is deprecated as of ggplot2 3.4.0.
ℹ Please use the `linewidth` argument instead.
This warning is displayed once every 8 hours.
Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
4: A numeric `legend.position` argument in `theme()` was deprecated in ggplot2 3.5.0.
ℹ Please use the `legend.position.inside` argument of `theme()` instead.
This warning is displayed once every 8 hours.
Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
>
> # Add labels to outliers using ggrepel
> p <- p + geom_text_repel(
+   data = subset(df, outlier),
+   aes(label = label),
+   box.padding = 0.5,
+   point.padding = 0.5,
+   force = 2,
+   max.overlaps = Inf,
+   size = 3.5
+ )
>
> # Add regression equation
> eq <- paste0("y = ", round(coef(lm_model)[2], 4), "x + ", round(coef(lm_model)[1], 4))
> p <- p + annotate("text", x = -Inf, y = Inf, label = eq, hjust = -1.1, vjust = 16, size = 4)
>
> # Extend the plot limits (less than before)
> x_range <- max(df$x) - min(df$x)
> y_range <- max(df$y) - min(df$y)
> p <- p + coord_cartesian(
+   xlim = c(min(df$x) - 0.01* x_range, max(df$x) + 0.01 * x_range),
+   ylim = c(min(df$y) - 0.01 * y_range, max(df$y) + 0.01 * y_range)
+ )
>
> # Add Pearson correlation to the plot
> cor_text <- paste("Pearson correlation:", round(cor_result$estimate, 4))
> p <- p + annotate("text", x = -Inf, y = Inf, label = cor_text, hjust = -0.8, vjust = 12.5, size = 4.5)
>
> # Display the plot
> print(p)
`geom_smooth()` using formula = 'y ~ x'
>
>
```