Interdependence between Drugs

```
library(BCor) # Install via remotes::install_github("jan-lukas-wermuth/BCor") if not done yet
```

```
## Warning: replacing previous import 'MASS::select' by 'dplyr::select' when
## loading 'BCor'

## Warning: replacing previous import 'dplyr::lag' by 'stats::lag' when loading
## 'BCor'

## Warning: replacing previous import 'dplyr::filter' by 'stats::filter' when
## loading 'BCor'
```

```
# Package names
packages <- c("tidyverse",
              "here",
              "mvtnorm",
              "readxl", # import from Excel
              "formatR", # 'double'-pipe
              "patchwork", # plot arrangement
              "Cairo") # save pdf (only necessary to have \phi properly illustrated with ggsave). This may

# Install packages
# Loop: checks if packages are already installed. If not, they will be installed.
# Otherwise, they will be updated.
for(i in packages){
  if(!require(i, character.only = T)){
    install.packages(i, dependencies = T)
    require(i, character.only = T)
  }
}
```

```
## Loading required package: tidyverse

## Warning: package 'ggplot2' was built under R version 4.3.3

## Warning: package 'purrr' was built under R version 4.3.3

## -- 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   3.5.2     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.4

## -- 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 become errors
## Loading required package: here
##
## here() starts at /Users/lukaswermuth/Library/CloudStorage/Dropbox/DimitriadisPohleWermuth/Binary Correla
##
## Loading required package: mvtnorm

## Warning: package 'mvtnorm' was built under R version 4.3.3
## Loading required package: readxl

## Warning: package 'readxl' was built under R version 4.3.3

## Loading required package: formatR
## Loading required package: patchwork
## Loading required package: Cairo

## Warning: package 'Cairo' was built under R version 4.3.3
```

Preparation work

Load data in terms of p, q and r. The structure of pqr_arr is the following: The first dimension marks the first drug and the second dimension the second drug. The third dimension marks p, q or r. p always refers to the relative consumption frequency of the first drug in the comparison and q to the relative consumption frequency of the second drug in the comparison. r denotes the joint relative consumption frequency.

```
load(here("application/data_pqr/pqr_arr.RData"))
```

Load functions:

1. comparison_binary: Function yielding a list containing Cole's C, Yule's Q and Phi.
2. plot_contour_matrix: Function creating heatmap with rows/columns ordered according to skewness of the marginal distribution.

```
# Insert variable1 and variable2 as strings, e.g. 'MET'
comparison_binary <- function(variable1, variable2) {
    phi <- as.numeric(Phi(pqr_arr[variable1, variable2, c("p",
        "q", "r")], alpha = FALSE))
    cole <- as.numeric(Cole(pqr_arr[variable1, variable2, c("p",
        "q", "r")], alpha = FALSE))
    yule <- as.numeric(YuleQ(pqr_arr[variable1, variable2, c("p",
        "q", "r")], alpha = FALSE))
    result <- list(Phi = phi, `Cole's C` = cole, `Yule's Q` = yule)
    return(result)
}
plot_contour_matrix <- function(mat, xlab, ylab, bins) {
    drug_d_sorted <- as.factor(c("ALC", "MAR", "CIG", "AMP",
        "TRQ", "NAR", "LSD", "PSY", "SED", "MDM", "COK", "CRA",
        "MET", "HER"))  # drugs sorted according to their marginal relative consumption frequencies
    drug_d_sorted_desc <- as.factor(rev(drug_d_sorted))

    data <- expand_grid(x = drug_d_sorted, y = drug_d_sorted_desc)

    z <- rep(NA, dim(mat)[1]^2)
    for (i in 1:dim(mat)[1]) {
        for (j in 1:dim(mat)[1]) {
            z[(i - 1) * dim(mat)[1] + j] <- mat[j, i]
        }
    }

    data <- mutate(data, z)

    labels <- rep("", bins + 1)

    addlabels <- seq(from = -1, to = 1, length.out = 11)

    addlabelsat <- seq(from = 1, to = bins + 1, length.out = 11)
    labels[addlabelsat] <- addlabels

    g <- ggplot(data, aes(x = fct_inorder(x), y = fct_inorder(y),
        fill = z)) + geom_tile() + scale_fill_stepsn(n.breaks = bins,
        colors = hcl.colors(20, "Spectral"), limits = c(-1, 1),
        labels = labels) + guides(fill = guide_colourbar(title = NULL,
        ticks = FALSE, barheight = 10)) + xlab(xlab) + ylab(ylab)
    g
    # 'Green-Brown' or 'Blue-Red 3' instead of 'Spectral'
    # are colorblind-safe, but for smaller ranges of values
    # do not distinguish as well
}
```

Create sorted (according to skewness) drug name vector in order to have the correlation (Cole's C, Yule's Q, Phi) matrices in the order in which the heatmap function needs it to be.

```
drug_d_sorted <- c("ALC", "MAR", "CIG", "AMP", "TRQ", "NAR",
    "LSD", "PSY", "SED", "MDM", "COK", "CRA", "MET", "HER")  # drugs sorted according to their marginal re
drug_d_sorted_desc <- rev(drug_d_sorted)
```

Create correlation matrices (Cole's C, Yule's Q, Phi):

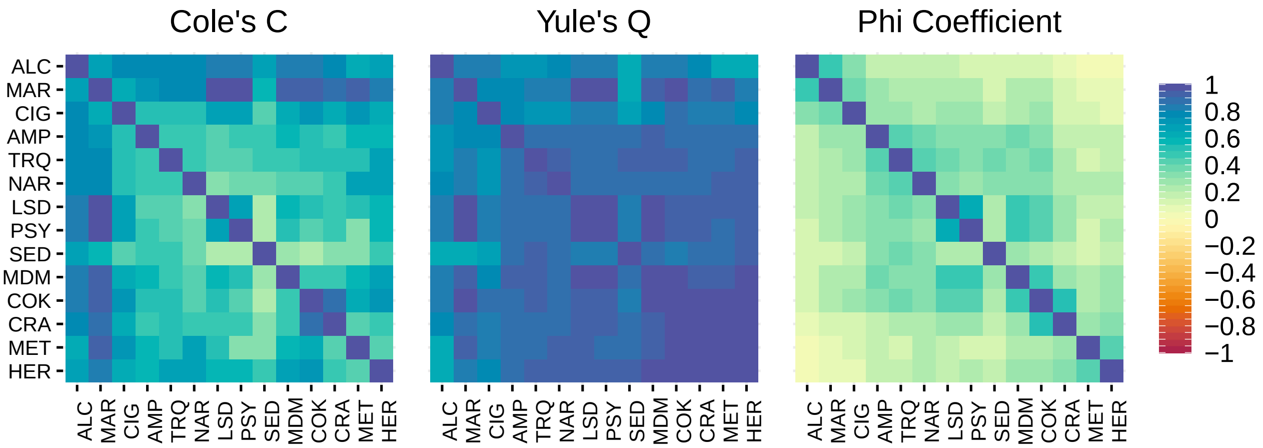
```
# Matrix 1: Phi
# ---------------------------------------------------------
df_phi <- matrix(NA, ncol = length(drug_d_sorted), nrow = length(drug_d_sorted))
colnames(df_phi) <- drug_d_sorted
rownames(df_phi) <- drug_d_sorted_desc
k <- 0

# Fill the data frame with the appropriate dependence
# measures for the appropriate comparison (sex vs.
# consumption of some drug)
for (i in drug_d_sorted_desc) {
    k <- k + 1
    l <- 0
    for (j in drug_d_sorted) {
        l <- l + 1
        df_phi[k, l] <- comparison_binary(i, j)[[1]]
    }
}

# Matrix 2: Cole's C
# ---------------------------------------------------------
df_cole <- matrix(NA, ncol = length(drug_d_sorted), nrow = length(drug_d_sorted))
colnames(df_cole) <- drug_d_sorted
rownames(df_cole) <- drug_d_sorted_desc
k <- 0

# Fill the data frame with the appropriate dependence
# measures for the appropriate comparison (sex vs.
# consumption of some drug)
for (i in drug_d_sorted_desc) {
    k <- k + 1
    l <- 0
    for (j in drug_d_sorted) {
        l <- l + 1
        df_cole[k, l] <- comparison_binary(i, j)[[2]]
```

```r
    }
}


# Matrix 3: Yule's Q
# ---------------------------------------------------------
df_yule <- matrix(NA, ncol = length(drug_d_sorted), nrow = length(drug_d_sorted))
colnames(df_yule) <- drug_d_sorted
rownames(df_yule) <- drug_d_sorted_desc
k <- 0

# Fill the data frame with the appropriate dependence
# measures for the appropriate comparison (sex vs.
# consumption of some drug)
for (i in drug_d_sorted_desc) {
    k <- k + 1
    l <- 0
    for (j in drug_d_sorted) {
        l <- l + 1
        df_yule[k, l] <- comparison_binary(i, j)[[3]]
    }
}
```

Including Plots

Equipped with those three matrices, we can produce the respective heatmaps:

```
adj_width <- 50
adj_height <- 96
size <- 14

plot_contour_matrix(df_cole, "", "", 40) + coord_fixed() + theme_minimal(base_size = size) +
    ggtitle("Cole's C") + theme(plot.title = element_text(hjust = 0.5)) +
    theme(axis.text.x = element_text(color = "black", angle = 90),
        axis.text.y = element_text(color = "black"), axis.ticks = element_line(color = "black"),
        legend.text = element_text(size = size)) + plot_contour_matrix(df_yule,
    "", "", 40) + coord_fixed() + theme_minimal(base_size = size) +
    ggtitle("Yule's Q") + theme(plot.title = element_text(hjust = 0.5)) +
    theme(axis.text.x = element_text(color = "black", angle = 90),
        axis.ticks.x = element_line(color = "black"), axis.text.y = element_blank(),
        axis.ticks.y = element_blank(), axis.title.y = element_blank(),
        legend.text = element_text(size = size)) + plot_contour_matrix(df_phi,
    "", "", 40) + coord_fixed() + theme_minimal(base_size = size) +
    ggtitle("Phi Coefficient") + theme(plot.title = element_text(hjust = 0.5)) +
    theme(axis.text.x = element_text(color = "black", angle = 90),
        axis.ticks.x = element_line(color = "black"), axis.text.y = element_blank(),
        axis.ticks.y = element_blank(), axis.title.y = element_blank(),
        legend.text = element_text(size = size)) + plot_layout(guides = "collect",
    ncol = 3)
```

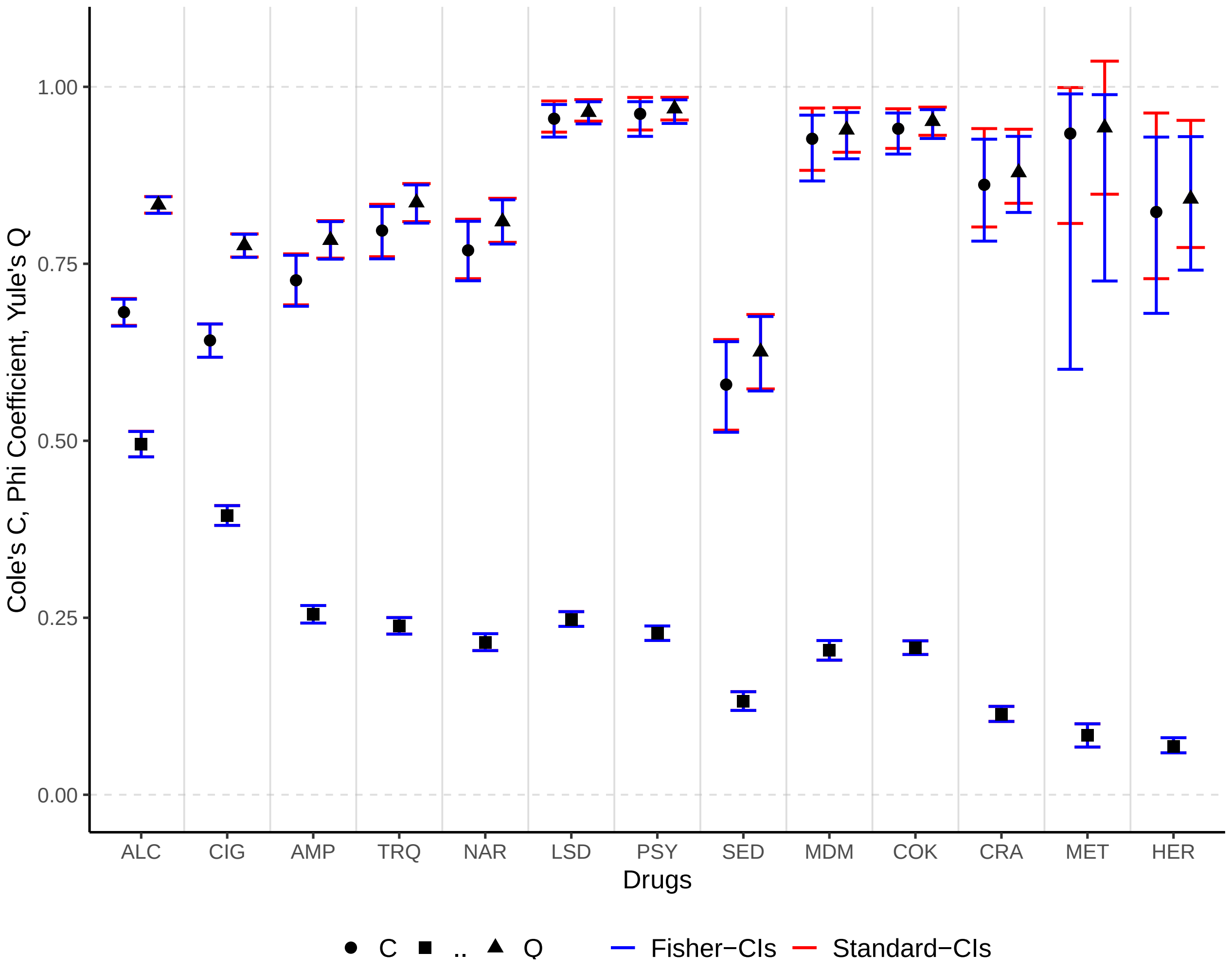


```
ggsave(filename = here("application/plots/drugs_comparison.pdf"),
    height = 200 - adj_height, width = (1 + sqrt(5))/2 * 200 -
        adj_width, device = "pdf", units = "mm")
```

The last piece of code computes the confidence intervals of Cole's C, Phi and Yule's Q for all comparisons involving marijuana and meth.

```
# Order drugs with respect to the frequency of use (except marijuana, which is our reference point)
  drugs_inference_sorted <- c("ALC", "CIG", "AMP", "TRQ", "NAR", "LSD", "PSY", "SED", "MDM", "COK", "CRA",
  drugs_inference_sorted_desc <- rev(drug_d_sorted)

marijuana_CI <- function(variable){ # function yielding all the estimates for Cole's C, Yule's Q and Phi a
  C_est <- Cole(pqr_arr["MAR", variable, c("p", "q", "r")], alpha = FALSE)[[1]]
  Phi_est <- Phi(pqr_arr["MAR", variable, c("p", "q", "r")], alpha = FALSE)[[1]]
  Q_est <- YuleQ(pqr_arr["MAR", variable, c("p", "q", "r")], alpha = FALSE)[[1]]
  marijuana_CI_C <- c(Cole(pqr_arr["MAR", variable, c("p", "q", "r")], Fisher = FALSE, alpha = 0.9, n = pq
  marijuana_CI_CZ <- c(Cole(pqr_arr["MAR", variable, c("p", "q", "r")], alpha = 0.9, n = pqr_arr["MAR", va
  marijuana_CI_Q <- c(YuleQ(pqr_arr["MAR", variable, c("p", "q", "r")], Fisher = FALSE, alpha = 0.9, n = p
  marijuana_CI_QZ <- c(YuleQ(pqr_arr["MAR", variable, c("p", "q", "r")], alpha = 0.9, n = pqr_arr["MAR", v
  marijuana_CI_phi <- c(Phi(pqr_arr["MAR", variable, c("p", "q", "r")], Fisher = FALSE, alpha = 0.9, n = p
  marijuana_CI_phiZ <- c(Phi(pqr_arr["MAR", variable, c("p", "q", "r")], alpha = 0.9, n = pqr_arr["MAR", v
  res <- dplyr::tribble(~C_est, ~Phi_est, ~Q_est, ~marijuana_CI_C, ~marijuana_CI_CZ, ~marijuana_CI_Q, ~mar
                          #--|--|--|--|--|--|--|--|--
                          C_est, Phi_est, Q_est, marijuana_CI_C, marijuana_CI_CZ, marijuana_CI_Q, marijuan
  return(res)
}

Q_est <- NA
C_est <- NA
Phi_est <- NA
CI_lower_Q <- NA
CI_upper_Q <- NA
CI_lower_QZ <- NA
CI_upper_QZ <- NA
CI_lower_C <- NA
CI_upper_C <- NA
CI_lower_CZ <- NA
CI_upper_CZ <- NA
CI_lower_phi <- NA
CI_upper_phi <- NA
CI_lower_phiZ <- NA
CI_upper_phiZ <- NA
for (i in drugs_inference_sorted) {
  Q_est <- c(Q_est, marijuana_CI(i)$Q_est)
}
for (i in drugs_inference_sorted) {
  C_est <- c(C_est, marijuana_CI(i)$C_est)
}
for (i in drugs_inference_sorted) {
  Phi_est <- c(Phi_est, marijuana_CI(i)$Phi_est)
}
for (i in drugs_inference_sorted) {
  CI_lower_Q <- c(CI_lower_Q, marijuana_CI(i)$marijuana_CI_Q[[1]][1])
}
for (i in drugs_inference_sorted) {
  CI_upper_Q <- c(CI_upper_Q, marijuana_CI(i)$marijuana_CI_Q[[1]][2])
}
for (i in drugs_inference_sorted) {
  CI_lower_QZ <- c(CI_lower_QZ, marijuana_CI(i)$marijuana_CI_QZ[[1]][1])
}
for (i in drugs_inference_sorted) {
  CI_upper_QZ <- c(CI_upper_QZ, marijuana_CI(i)$marijuana_CI_QZ[[1]][2])
}
for (i in drugs_inference_sorted) {
  CI_lower_C <- c(CI_lower_C, marijuana_CI(i)$marijuana_CI_C[[1]][1])
}
for (i in drugs_inference_sorted) {
  CI_upper_C <- c(CI_upper_C, marijuana_CI(i)$marijuana_CI_C[[1]][2])
}
for (i in drugs_inference_sorted) {
  CI_lower_CZ <- c(CI_lower_CZ, marijuana_CI(i)$marijuana_CI_CZ[[1]][1])
}
for (i in drugs_inference_sorted) {
  CI_upper_CZ <- c(CI_upper_CZ, marijuana_CI(i)$marijuana_CI_CZ[[1]][2])
}
for (i in drugs_inference_sorted) {
  CI_lower_phi <- c(CI_lower_phi, marijuana_CI(i)$marijuana_CI_phi[[1]][1])
}
for (i in drugs_inference_sorted) {
  CI_upper_phi <- c(CI_upper_phi, marijuana_CI(i)$marijuana_CI_phi[[1]][2])
}
for (i in drugs_inference_sorted) {
  CI_lower_phiZ <- c(CI_lower_phiZ, marijuana_CI(i)$marijuana_CI_phiZ[[1]][1])
}
for (i in drugs_inference_sorted) {
  CI_upper_phiZ <- c(CI_upper_phiZ, marijuana_CI(i)$marijuana_CI_phiZ[[1]][2])
}
# Need to omit the first entry (NA)
Q_est <- as.numeric(na.omit(Q_est))
C_est <- as.numeric(na.omit(C_est))
Phi_est <- as.numeric(na.omit(Phi_est))
CI_lower_Q <- as.numeric(na.omit(CI_lower_Q))
CI_upper_Q <- as.numeric(na.omit(CI_upper_Q))
CI_lower_QZ <- as.numeric(na.omit(CI_lower_QZ))
CI_upper_QZ <- as.numeric(na.omit(CI_upper_QZ))
CI_lower_C <- as.numeric(na.omit(CI_lower_C))
CI_upper_C <- as.numeric(na.omit(CI_upper_C))
CI_lower_CZ <- as.numeric(na.omit(CI_lower_CZ))
CI_upper_CZ <- as.numeric(na.omit(CI_upper_CZ))
CI_lower_phi <- as.numeric(na.omit(CI_lower_phi))
CI_upper_phi <- as.numeric(na.omit(CI_upper_phi))
CI_lower_phiZ <- as.numeric(na.omit(CI_lower_phiZ))
CI_upper_phiZ <- as.numeric(na.omit(CI_upper_phiZ))

data_Q <- data.frame(drugs = fct_inorder(drugs_inference_sorted),
                   Q_est = Q_est,
                   C_est = C_est,
                   Phi_est = Phi_est,
                   CI_lower_Q = CI_lower_Q,
                   CI_upper_Q = CI_upper_Q,
                   CI_lower_QZ = CI_lower_QZ,
                   CI_upper_QZ = CI_upper_QZ,
                   CI_lower_C = CI_lower_C,
                   CI_upper_C = CI_upper_C,
                   CI_lower_CZ = CI_lower_CZ,
                   CI_upper_CZ = CI_upper_CZ,
                   CI_lower_phi = CI_lower_phi,
                   CI_upper_phi = CI_upper_phi,
                   CI_lower_phiZ = CI_lower_phiZ,
                   CI_upper_phiZ = CI_upper_phiZ)
options(ggplot2.discrete.colour= c("blue", "red"))
ggplot(data_Q, aes(drugs, Q_est)) +
    geom_errorbar(aes(ymin = CI_lower_Q, ymax = CI_upper_Q, color = "Standard-CIs"), linewidth = 0.8, widt
    geom_errorbar(aes(ymin = CI_lower_QZ, ymax = CI_upper_QZ, color = "Fisher-CIs"), linewidth = 0.8, widt
    geom_errorbar(aes(ymin = CI_lower_C, ymax = CI_upper_C, color = "Standard-CIs"), linewidth = 0.8, widt
    geom_errorbar(aes(ymin = CI_lower_CZ, ymax = CI_upper_CZ, color = "Fisher-CIs"), linewidth = 0.8, widt
    geom_errorbar(aes(ymin = CI_lower_phi, ymax = CI_upper_phi, color = "Standard-CIs"), linewidth = 0.8,
    geom_errorbar(aes(ymin = CI_lower_phiZ, ymax = CI_upper_phiZ, color = "Fisher-CIs"), linewidth = 0.8,
    geom_point(aes(drugs, Q_est, shape = "Q"), size = 3, position = position_nudge(x = 0.2)) +
    geom_point(aes(drugs, C_est, shape = "C"), size = 3, position = position_nudge(x = -0.2)) +
    geom_point(aes(drugs, Phi_est, shape = "\u03D5"), size = 3) +
    labs(x = "Drugs",
         y = "Cole's C, Phi Coefficient, Yule's Q") +
    geom_hline(yintercept = c(0,1), linetype = "dashed", color = "grey", alpha = 0.5) +
    geom_vline(xintercept = c(1.5, 2.5, 3.5, 4.5, 5.5, 6.5, 7.5, 8.5, 9.5, 10.5, 11.5, 12.5), color = "gre
    theme_classic(base_size = 15) +
    theme(legend.text = element_text(size=15), legend.position = "bottom", axis.title=element_text(size=15
    scale_shape_manual(values = c(15, 16, 17)) +
    scale_colour_discrete(name = "") +
    scale_shape_discrete(name = "", breaks = c("C", "\u03D5", "Q")) +
    ylim(0,1.06)
```

```
## Scale for shape is already present.
## Adding another scale for shape, which will replace the existing scale.

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <cf>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <95>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <cf>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <95>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <cf>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <95>
```



```
ggsave(filename = here("application/plots/QCPhi_application_inference.pdf"), height = 200, width = (1+sqrt
```

```
# Same figure as above but only with Fisher-CIs
options(ggplot2.discrete.colour = c("blue", "red", "green4"))
ggplot(data_Q, aes(drugs, Q_est)) + geom_errorbar(aes(ymin = CI_lower_QZ,
    ymax = CI_upper_QZ), color = "black", linewidth = 0.8, width = 0.3,
    position = position_nudge(x = 0.2)) + geom_errorbar(aes(ymin = CI_lower_CZ,
    ymax = CI_upper_CZ), color = "black", linewidth = 0.8, width = 0.3,
    position = position_nudge(x = -0.2)) + geom_errorbar(aes(ymin = CI_lower_phiZ,
    ymax = CI_upper_phiZ), color = "black", linewidth = 0.8,
    width = 0.3) + geom_point(aes(drugs, Q_est, shape = "Q",
    color = "Q"), size = 3, position = position_nudge(x = 0.2)) +
    geom_point(aes(drugs, C_est, shape = "C", color = "C"), size = 3,
        position = position_nudge(x = -0.2)) + geom_point(aes(drugs,
    Phi_est, shape = " ", color = " "), size = 3) + labs(x = "Drugs",
    y = "Cole's C, Phi Coefficient, Yule's Q") + geom_hline(yintercept = c(0,
    1), linetype = "dashed", color = "grey", alpha = 0.5) + geom_vline(xintercept = c(1.5,
    2.5, 3.5, 4.5, 5.5, 6.5, 7.5, 8.5, 9.5, 10.5, 11.5, 12.5),
    color = "grey", alpha = 0.5) + theme_classic(base_size = 15) +
    theme(legend.text = element_text(size = 15), legend.position = "bottom",
        axis.title = element_text(size = 15)) + scale_colour_manual(values = c("blue",
    "red", "yellow")) + scale_shape_manual(values = c(15, 16,
    17)) + scale_colour_discrete(name = "", breaks = c("C", " ",
    "Q")) + scale_shape_discrete(name = "", breaks = c("C", " ",
    "Q")) + ylim(0, 1.06)
```

```
## Scale for colour is already present.
## Adding another scale for colour, which will replace the existing scale.
## Scale for shape is already present.
## Adding another scale for shape, which will replace the existing scale.
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <cf>
```

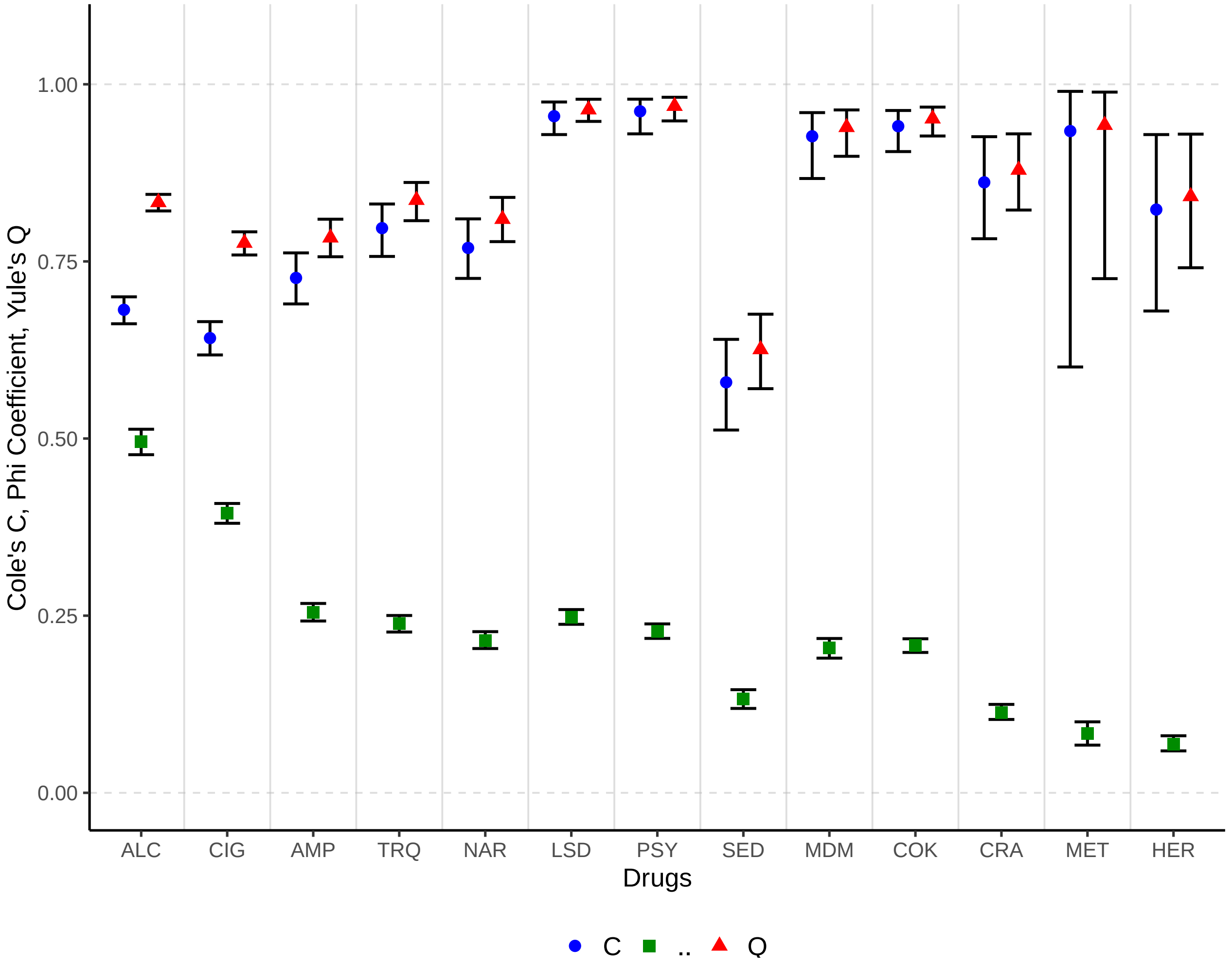
```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <95>
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <cf>
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <95>
```

```
## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <cf>
```

```
## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <95>
```



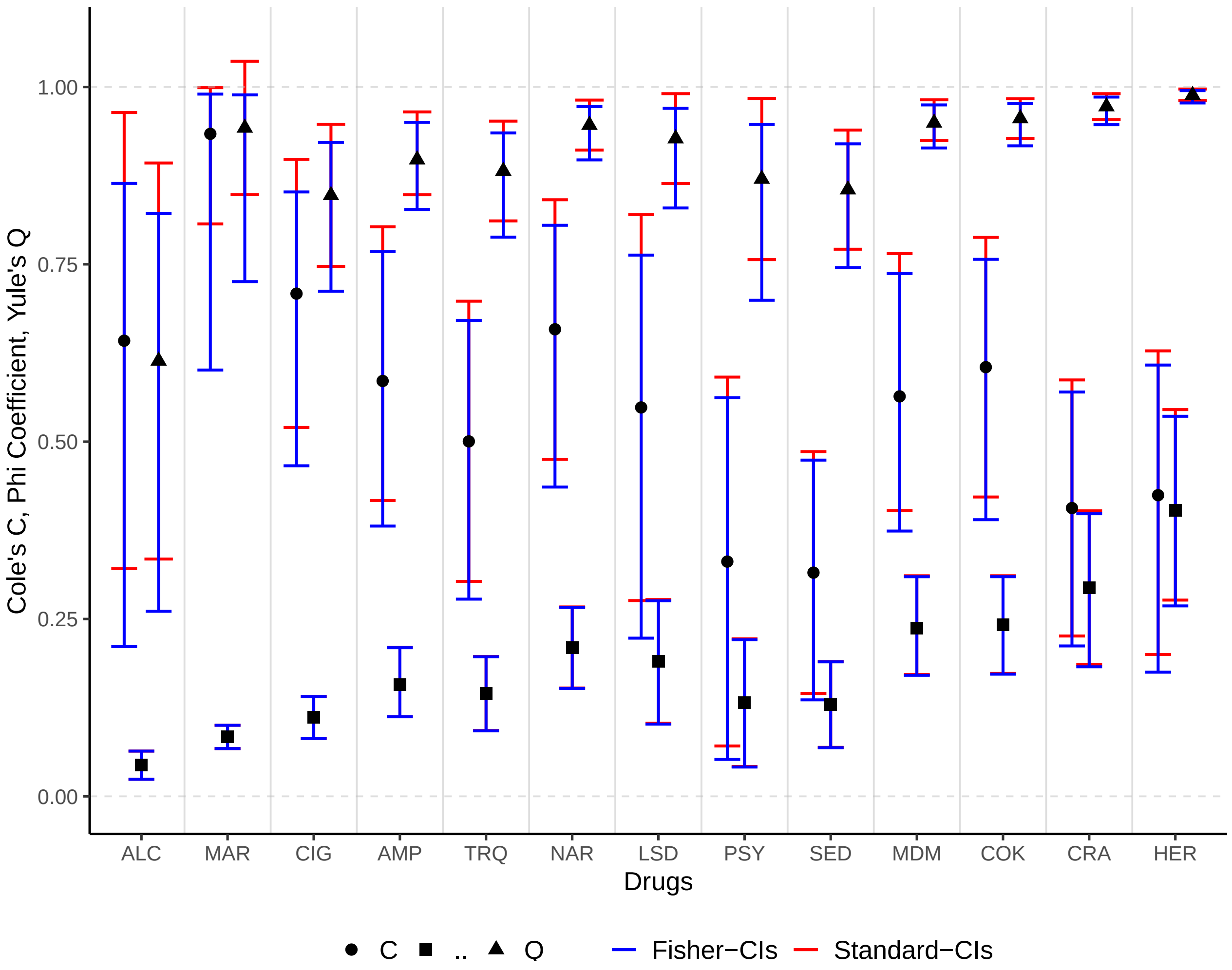
```
ggsave(filename = here("application/plots/QCPhi_application_inference_final.pdf"),
    height = 200, width = (1 + sqrt(5))/2 * 200, device = cairo_pdf,
    units = "mm")
```

```
# Order drugs with respect to the frequency of use (except meth, which is our reference point)
  drugs_inference_sorted <- c("ALC", "MAR", "CIG", "AMP", "TRQ", "NAR", "LSD", "PSY", "SED", "MDM", "COK",
```

```r
  drugs_inference_sorted_desc <- rev(drug_d_sorted)

meth_CI <- function(variable){ # function yielding all the estimates for Cole's C, Yule's Q and Phi as wel
  C_est <- Cole(pqr_arr["MET", variable, c("p", "q", "r")], alpha = FALSE)[[1]]
  Phi_est <- Phi(pqr_arr["MET", variable, c("p", "q", "r")], alpha = FALSE)[[1]]
  Q_est <- YuleQ(pqr_arr["MET", variable, c("p", "q", "r")], alpha = FALSE)[[1]]
  meth_CI_C <- c(Cole(pqr_arr["MET", variable, c("p", "q", "r")], Fisher = FALSE, alpha = 0.9, n = pqr_arr
  meth_CI_CZ <- c(Cole(pqr_arr["MET", variable, c("p", "q", "r")], alpha = 0.9, n = pqr_arr["MET", variabl
  meth_CI_Q <- c(YuleQ(pqr_arr["MET", variable, c("p", "q", "r")], Fisher = FALSE, alpha = 0.9, n = pqr_ar
  meth_CI_QZ <- c(YuleQ(pqr_arr["MET", variable, c("p", "q", "r")], alpha = 0.9, n = pqr_arr["MET", variab
  meth_CI_phi <- c(Phi(pqr_arr["MET", variable, c("p", "q", "r")], Fisher = FALSE, alpha = 0.9, n = pqr_ar
  meth_CI_phiZ <- c(Phi(pqr_arr["MET", variable, c("p", "q", "r")], alpha = 0.9, n = pqr_arr["MET", variab
  res <- dplyr::tribble(~C_est, ~Phi_est, ~Q_est, ~meth_CI_C, ~meth_CI_CZ, ~meth_CI_Q, ~meth_CI_QZ, ~meth_
                        #--/--/--/--/--/--/--/--/--
                        C_est, Phi_est, Q_est, meth_CI_C, meth_CI_CZ, meth_CI_Q, meth_CI_QZ, meth_CI_phi
  return(res)
}

Q_est <- NA
C_est <- NA
Phi_est <- NA
CI_lower_Q <- NA
CI_upper_Q <- NA
CI_lower_QZ <- NA
CI_upper_QZ <- NA
CI_lower_C <- NA
CI_upper_C <- NA
CI_lower_CZ <- NA
CI_upper_CZ <- NA
CI_lower_phi <- NA
CI_upper_phi <- NA
CI_lower_phiZ <- NA
CI_upper_phiZ <- NA
for (i in drugs_inference_sorted) {
  Q_est <- c(Q_est, meth_CI(i)$Q_est)
}
for (i in drugs_inference_sorted) {
  C_est <- c(C_est, meth_CI(i)$C_est)
}
for (i in drugs_inference_sorted) {
  Phi_est <- c(Phi_est, meth_CI(i)$Phi_est)
}
for (i in drugs_inference_sorted) {
  CI_lower_Q <- c(CI_lower_Q, meth_CI(i)$meth_CI_Q[[1]][1])
}
for (i in drugs_inference_sorted) {
  CI_upper_Q <- c(CI_upper_Q, meth_CI(i)$meth_CI_Q[[1]][2])
}
for (i in drugs_inference_sorted) {
  CI_lower_QZ <- c(CI_lower_QZ, meth_CI(i)$meth_CI_QZ[[1]][1])
}
for (i in drugs_inference_sorted) {
  CI_upper_QZ <- c(CI_upper_QZ, meth_CI(i)$meth_CI_QZ[[1]][2])
}
for (i in drugs_inference_sorted) {
  CI_lower_C <- c(CI_lower_C, meth_CI(i)$meth_CI_C[[1]][1])
}
for (i in drugs_inference_sorted) {
  CI_upper_C <- c(CI_upper_C, meth_CI(i)$meth_CI_C[[1]][2])
}
for (i in drugs_inference_sorted) {
  CI_lower_CZ <- c(CI_lower_CZ, meth_CI(i)$meth_CI_CZ[[1]][1])
}
for (i in drugs_inference_sorted) {
  CI_upper_CZ <- c(CI_upper_CZ, meth_CI(i)$meth_CI_CZ[[1]][2])
}
for (i in drugs_inference_sorted) {
  CI_lower_phi <- c(CI_lower_phi, meth_CI(i)$meth_CI_phi[[1]][1])
}
for (i in drugs_inference_sorted) {
  CI_upper_phi <- c(CI_upper_phi, meth_CI(i)$meth_CI_phi[[1]][2])
}
for (i in drugs_inference_sorted) {
  CI_lower_phiZ <- c(CI_lower_phiZ, meth_CI(i)$meth_CI_phiZ[[1]][1])
}
for (i in drugs_inference_sorted) {
  CI_upper_phiZ <- c(CI_upper_phiZ, meth_CI(i)$meth_CI_phiZ[[1]][2])
}
# Need to omit the first entry (NA)
Q_est <- as.numeric(na.omit(Q_est))
C_est <- as.numeric(na.omit(C_est))
Phi_est <- as.numeric(na.omit(Phi_est))
CI_lower_Q <- as.numeric(na.omit(CI_lower_Q))
CI_upper_Q <- as.numeric(na.omit(CI_upper_Q))
CI_lower_QZ <- as.numeric(na.omit(CI_lower_QZ))
CI_upper_QZ <- as.numeric(na.omit(CI_upper_QZ))
CI_lower_C <- as.numeric(na.omit(CI_lower_C))
CI_upper_C <- as.numeric(na.omit(CI_upper_C))
CI_lower_CZ <- as.numeric(na.omit(CI_lower_CZ))
CI_upper_CZ <- as.numeric(na.omit(CI_upper_CZ))
CI_lower_phi <- as.numeric(na.omit(CI_lower_phi))
CI_upper_phi <- as.numeric(na.omit(CI_upper_phi))
CI_lower_phiZ <- as.numeric(na.omit(CI_lower_phiZ))
CI_upper_phiZ <- as.numeric(na.omit(CI_upper_phiZ))

data_Q <- data.frame(drugs = fct_inorder(drugs_inference_sorted),
                     Q_est = Q_est,
                     C_est = C_est,
                     Phi_est = Phi_est,
                     CI_lower_Q = CI_lower_Q,
                     CI_upper_Q = CI_upper_Q,
                     CI_lower_QZ = CI_lower_QZ,
                     CI_upper_QZ = CI_upper_QZ,
                     CI_lower_C = CI_lower_C,
                     CI_upper_C = CI_upper_C,
                     CI_lower_CZ = CI_lower_CZ,
                     CI_upper_CZ = CI_upper_CZ,
                     CI_lower_phi = CI_lower_phi,
                     CI_upper_phi = CI_upper_phi,
                     CI_lower_phiZ = CI_lower_phiZ,
                     CI_upper_phiZ = CI_upper_phiZ)
options(ggplot2.discrete.colour= c("blue", "red"))
```

```
ggplot(data_Q, aes(drugs, Q_est)) +
    geom_errorbar(aes(ymin = CI_lower_Q, ymax = CI_upper_Q, color = "Standard-CIs"), linewidth = 0.8, widt
    geom_errorbar(aes(ymin = CI_lower_QZ, ymax = CI_upper_QZ, color = "Fisher-CIs"), linewidth = 0.8, widt
    geom_errorbar(aes(ymin = CI_lower_C, ymax = CI_upper_C, color = "Standard-CIs"), linewidth = 0.8, widt
    geom_errorbar(aes(ymin = CI_lower_CZ, ymax = CI_upper_CZ, color = "Fisher-CIs"), linewidth = 0.8, widt
    geom_errorbar(aes(ymin = CI_lower_phi, ymax = CI_upper_phi, color = "Standard-CIs"), linewidth = 0.8,
    geom_errorbar(aes(ymin = CI_lower_phiZ, ymax = CI_upper_phiZ, color = "Fisher-CIs"), linewidth = 0.8,
    geom_point(aes(drugs, Q_est, shape = "Q"), size = 3, position = position_nudge(x = 0.2)) +
    geom_point(aes(drugs, C_est, shape = "C"), size = 3, position = position_nudge(x = -0.2)) +
    geom_point(aes(drugs, Phi_est, shape = "\u03D5"), size = 3) +
    labs(x = "Drugs",
         y = "Cole's C, Phi Coefficient, Yule's Q") +
    geom_hline(yintercept = c(0,1), linetype = "dashed", color = "grey", alpha = 0.5) +
    geom_vline(xintercept = c(1.5, 2.5, 3.5, 4.5, 5.5, 6.5, 7.5, 8.5, 9.5, 10.5, 11.5, 12.5), color = "gre
    theme_classic(base_size = 15) +
    theme(legend.text = element_text(size=15), legend.position = "bottom", axis.title=element_text(size=15
    scale_colour_manual(values = c("blue", "red")) +
    scale_shape_manual(values = c(15, 16, 17)) +
    scale_colour_discrete(name = "") +
    scale_shape_discrete(name = "", breaks = c("C", "\u03D5", "Q")) +
    ylim(0,1.06)
```

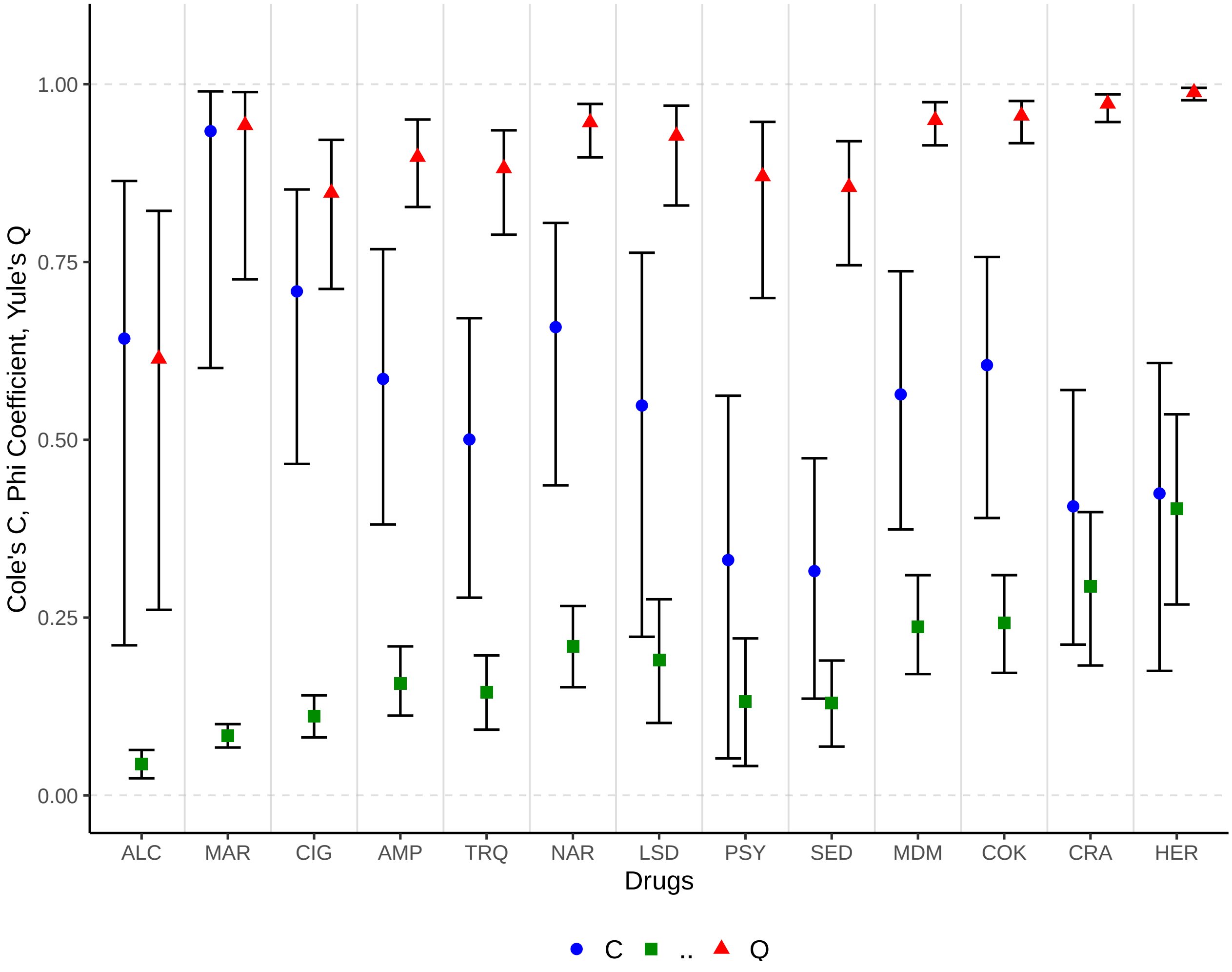
```
## Scale for colour is already present.
## Adding another scale for colour, which will replace the existing scale.
## Scale for shape is already present.
## Adding another scale for shape, which will replace the existing scale.

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <cf>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <95>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <cf>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <95>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <cf>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <95>
```



```
ggsave(filename = here("application/plots/QCPhi_application_inference_meth.pdf"), height = 200, width = (1
```

```
# Same figure as above but only with Fisher-CIs
options(ggplot2.discrete.colour = c("blue", "red", "green4"))
ggplot(data_Q, aes(drugs, Q_est)) + geom_errorbar(aes(ymin = CI_lower_QZ,
    ymax = CI_upper_QZ), color = "black", linewidth = 0.7, width = 0.3,
    position = position_nudge(x = 0.2)) + geom_errorbar(aes(ymin = CI_lower_CZ,
    ymax = CI_upper_CZ), color = "black", linewidth = 0.7, width = 0.3,
    position = position_nudge(x = -0.2)) + geom_errorbar(aes(ymin = CI_lower_phiZ,
    ymax = CI_upper_phiZ), color = "black", linewidth = 0.7,
    width = 0.3) + geom_point(aes(drugs, Q_est, shape = "Q",
    color = "Q"), size = 3, position = position_nudge(x = 0.2)) +
    geom_point(aes(drugs, C_est, shape = "C", color = "C"), size = 3,
        position = position_nudge(x = -0.2)) + geom_point(aes(drugs,
    Phi_est, shape = " ", color = " "), size = 3) + labs(x = "Drugs",
    y = "Cole's C, Phi Coefficient, Yule's Q") + geom_hline(yintercept = c(0,
    1), linetype = "dashed", color = "grey", alpha = 0.5) + geom_vline(xintercept = c(1.5,
    2.5, 3.5, 4.5, 5.5, 6.5, 7.5, 8.5, 9.5, 10.5, 11.5, 12.5),
    color = "grey", alpha = 0.5) + theme_classic(base_size = 15) +
    theme(legend.text = element_text(size = 15), legend.position = "bottom",
        axis.title = element_text(size = 15)) + scale_colour_manual(values = c("blue",
    "red")) + scale_shape_manual(values = c(15, 16, 17)) + scale_colour_discrete(name = "",
    breaks = c("C", " ", "Q")) + scale_shape_discrete(name = "",
    breaks = c("C", " ", "Q")) + ylim(0, 1.06)
```

```
## Scale for colour is already present.
## Adding another scale for colour, which will replace the existing scale.
## Scale for shape is already present.
## Adding another scale for shape, which will replace the existing scale.

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <cf>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <95>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <cf>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <95>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <cf>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <95>
```



```
ggsave(filename = here("application/plots/QCPhi_application_inference_meth_final.pdf"),
    height = 200, width = (1 + sqrt(5))/2 * 200, device = cairo_pdf,
    units = "mm")
```