Heatmaps Phi Bounds and Cole's C vs. Yule's Q vs. Phi

Function: Compare coefficient1 with coefficient2 by using a heatmap

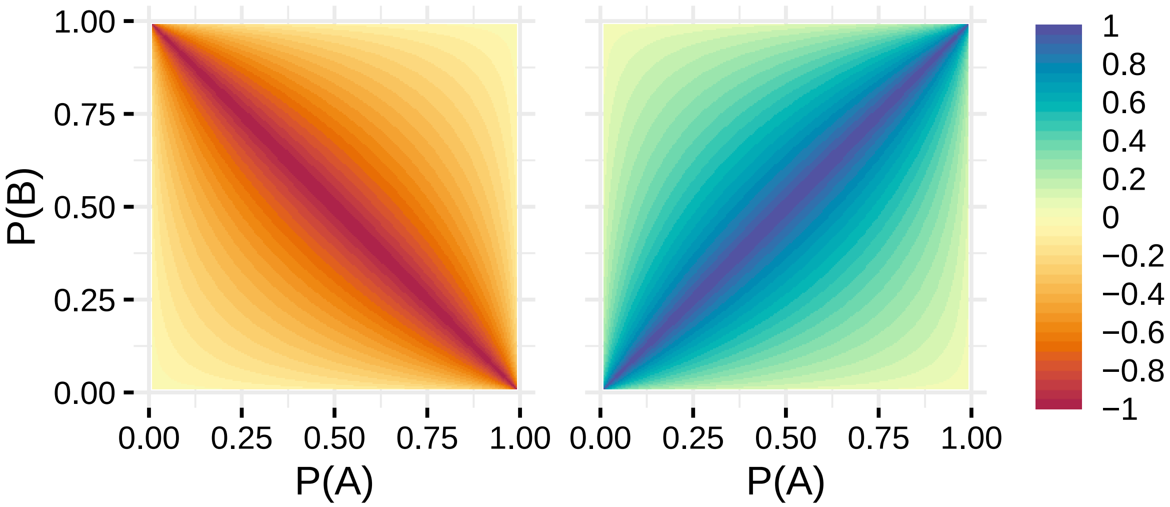
- coefficient1: Coefficient that gets evaluated, i.e. whose values get displayed in the heatmap
- coefficient2.inv: Coefficient that gets fixed and used to create contingency tables for varying marginal distributions
- level: Level at which coefficient2 gets fixed
- xlim, ylim: range of PA and PB as a vector. Recommendation: use c(0.01, 0.99) but don't include 0 or 1. This will produce errors.
- xlab, ylab: axis description with strings
- PA: Marginal probability for event A (row 1 in contingency table)
- PB: Marginal probability for event B (column 1 in contingency table)

```
plot_contour_binary <- function(coefficient1, coefficient2.inv,
    level, xlim, ylim, grid, xlab, ylab, bins) {
    # function yielding the value for coefficient1 given
    # the contingency table which is determined by
    # coefficient2.inv, PA and PB
    f <- function(coefficient1, coefficient2.inv, PA, PB) {
        as.numeric(coefficient1(coefficient2.inv(level, c(PA,
            1 - PA, PB, 1 - PB)), alpha = FALSE))
    }

    PA <- seq(xlim[1], xlim[2], length.out = grid)
    PB <- seq(ylim[1], ylim[2], length.out = grid)
    data <- expand_grid(x = PA, y = PB)
    z <- rep(NA, grid^2)
    for (i in 1:grid) {
        for (j in 1:grid) {
            z[(i - 1) * grid + j] <- f(coefficient1, coefficient2.inv,
                PA[i], PB[j])
        }
    }
    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 = x, y = 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)

    # 'Green-Brown' or 'Blue-Red 3' instead of 'Spectral'
    # are colorblind-safe, but for smaller ranges of values
    # do not distinguish as well

    g
}
plot_contour_trivial <- function(level, xlim, ylim, grid, xlab,
    ylab, bins) {
    # function yielding the value for coefficient1 given
    # the contingency table which is determined by
    # coefficient2.inv, PA and PB
    f <- function(PA, PB) {
        level
    }

    PA <- seq(xlim[1], xlim[2], length.out = grid)
    PB <- seq(ylim[1], ylim[2], length.out = grid)
    data <- expand_grid(x = PA, y = PB)
    z <- rep(NA, grid^2)
    for (i in 1:grid) {
        for (j in 1:grid) {
            z[(i - 1) * grid + j] <- f(PA[i], PB[j])
        }
    }
    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 = x, y = 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)

    # 'Green-Brown' or 'Blue-Red 3' instead of 'Spectral'
    # are colorblind-safe, but for smaller ranges of values
    # do not distinguish as well

    g
}
```

Including Plots

Bounds for Phi

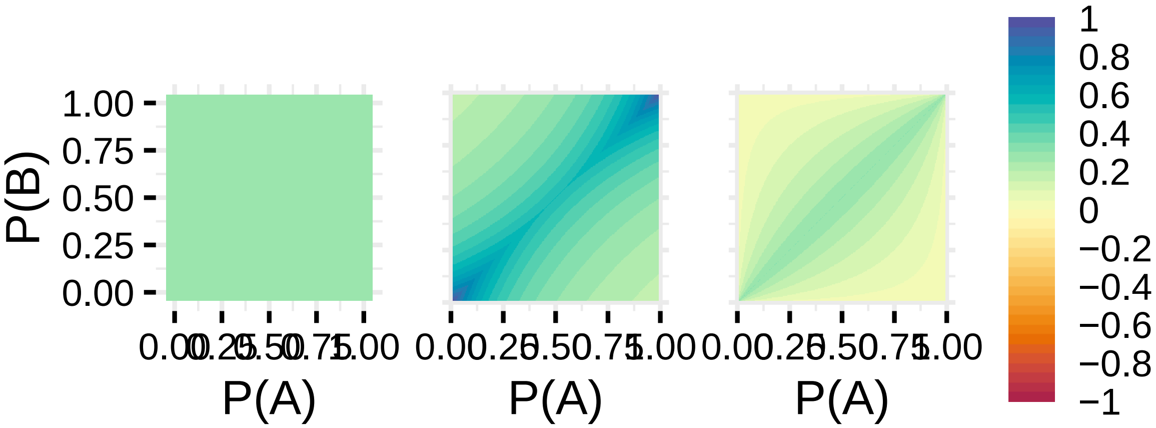
```
gridsize <- 250
plot_contour_binary(Phi, Cole.inv, -1, c(0.01, 0.99), c(0.01,
    0.99), gridsize, "P(A)", "P(B)", 40) + coord_fixed() + theme_minimal(base_size = 15) +
    theme(axis.text.x = element_text(color = "black"), axis.text.y = element_text(color = "black"),
        axis.ticks = element_line(color = "black")) + plot_contour_binary(Phi,
    Cole.inv, 1, c(0.01, 0.99), c(0.01, 0.99), gridsize, "P(A)",
    "P(B)", 40) + coord_fixed() + theme_minimal(base_size = 15) +
    theme(axis.text.x = element_text(color = "black"), axis.ticks.x = element_line(color = "black"),
        axis.text.y = element_blank(), axis.ticks.y = element_blank(),
        axis.title.y = element_blank()) + plot_layout(guides = "collect",
    ncol = 2)
```



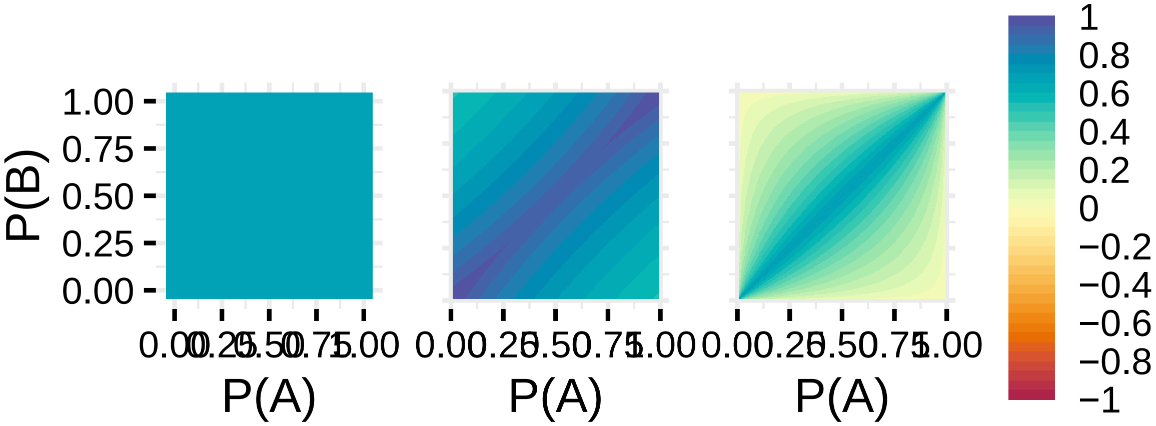
```
ggsave(filename = "~/Dropbox/DimitriadisPohleWermuth/Binary Correlation/replication_BCor/application/pl
    width = 200, height = 90, device = "pdf", units = "mm")
```

Heatmaps with Cole as the coefficient to be set and comparison with Cole, YuleQ and Phi

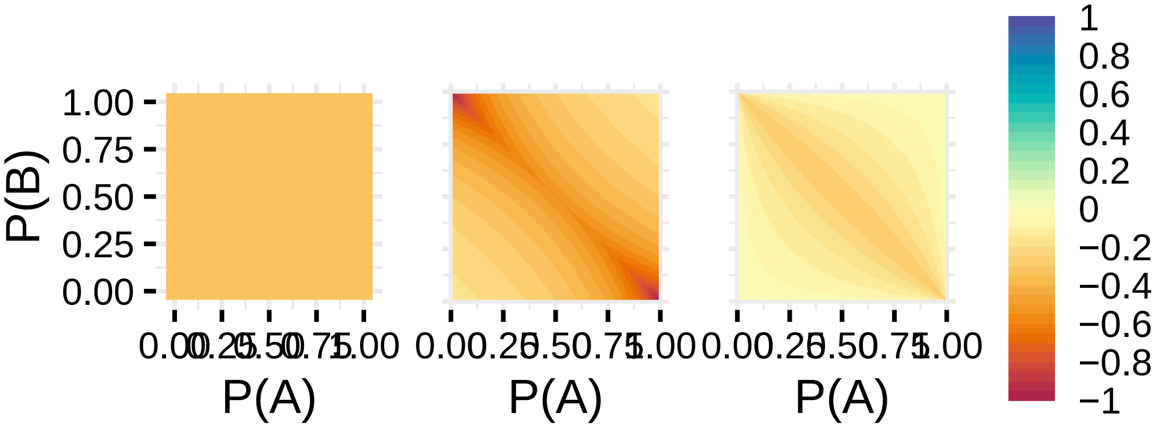
```
gridsize <- 250
adj_width <- 50
adj_height <- 108
base_size <- 18
plot_contour_trivial(0.3, c(0.01, 0.99), c(0.01, 0.99), 10, "P(A)",
    "P(B)", 40) + coord_fixed() + theme_minimal(base_size = base_size) +
    theme(axis.text.x = element_text(color = "black"), axis.text.y = element_text(color = "black"),
        axis.ticks = element_line(color = "black")) + plot_contour_binary(YuleQ,
    Cole.inv, 0.3, c(0.01, 0.99), c(0.01, 0.99), gridsize, "P(A)",
    "P(B)", 40) + coord_fixed() + theme_minimal(base_size = base_size) +
    theme(axis.text.x = element_text(color = "black"), axis.ticks.x = element_line(color = "black"),
        axis.text.y = element_blank(), axis.ticks.y = element_blank(),
        axis.title.y = element_blank()) + plot_contour_binary(Phi,
    Cole.inv, 0.3, c(0.01, 0.99), c(0.01, 0.99), gridsize, "P(A)",
    "P(B)", 40) + coord_fixed() + theme_minimal(base_size = base_size) +
    theme(axis.text.x = element_text(color = "black"), axis.ticks.x = element_line(color = "black"),
        axis.text.y = element_blank(), axis.ticks.y = element_blank(),
        axis.title.y = element_blank()) + plot_layout(guides = "collect",
    ncol = 3)
```



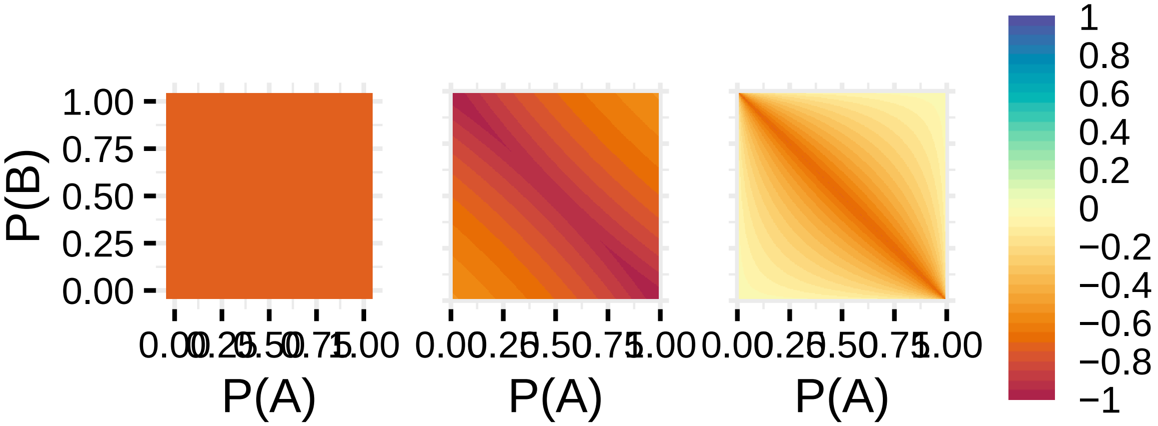
```
ggsave(filename = "~/Dropbox/DimitriadisPohleWermuth/Binary Correlation/replication_BCor/application/pl
    height = 200 - adj_height, width = (1 + sqrt(5))/2 * 200 -
        adj_width, device = "pdf", units = "mm")

plot_contour_trivial(0.7, c(0.01, 0.99), c(0.01, 0.99), 10, "P(A)",
    "P(B)", 40) + coord_fixed() + theme_minimal(base_size = base_size) +
    theme(axis.text.x = element_text(color = "black"), axis.text.y = element_text(color = "black"),
        axis.ticks = element_line(color = "black")) + plot_contour_binary(YuleQ,
    Cole.inv, 0.7, c(0.01, 0.99), c(0.01, 0.99), gridsize, "P(A)",
    "P(B)", 40) + coord_fixed() + theme_minimal(base_size = base_size) +
    theme(axis.text.x = element_text(color = "black"), axis.ticks.x = element_line(color = "black"),
        axis.text.y = element_blank(), axis.ticks.y = element_blank(),
        axis.title.y = element_blank()) + plot_contour_binary(Phi,
    Cole.inv, 0.7, c(0.01, 0.99), c(0.01, 0.99), gridsize, "P(A)",
    "P(B)", 40) + coord_fixed() + theme_minimal(base_size = base_size) +
    theme(axis.text.x = element_text(color = "black"), axis.ticks.x = element_line(color = "black"),
        axis.text.y = element_blank(), axis.ticks.y = element_blank(),
        axis.title.y = element_blank()) + plot_layout(guides = "collect",
    ncol = 3)
```



```
ggsave(filename = "~/Dropbox/DimitriadisPohleWermuth/Binary Correlation/replication_BCor/application/pl
    height = 200 - adj_height, width = (1 + sqrt(5))/2 * 200 -
        adj_width, device = "pdf", units = "mm")

plot_contour_trivial(-0.3, c(0.01, 0.99), c(0.01, 0.99), 10,
    "P(A)", "P(B)", 40) + coord_fixed() + theme_minimal(base_size = base_size) +
    theme(axis.text.x = element_text(color = "black"), axis.text.y = element_text(color = "black"),
        axis.ticks = element_line(color = "black")) + plot_contour_binary(YuleQ,
    Cole.inv, -0.3, c(0.01, 0.99), c(0.01, 0.99), gridsize, "P(A)",
    "P(B)", 40) + coord_fixed() + theme_minimal(base_size = base_size) +
    theme(axis.text.x = element_text(color = "black"), axis.ticks.x = element_line(color = "black"),
        axis.text.y = element_blank(), axis.ticks.y = element_blank(),
        axis.title.y = element_blank()) + plot_contour_binary(Phi,
    Cole.inv, -0.3, c(0.01, 0.99), c(0.01, 0.99), gridsize, "P(A)",
    "P(B)", 40) + coord_fixed() + theme_minimal(base_size = base_size) +
    theme(axis.text.x = element_text(color = "black"), axis.ticks.x = element_line(color = "black"),
        axis.text.y = element_blank(), axis.ticks.y = element_blank(),
        axis.title.y = element_blank()) + plot_layout(guides = "collect",
    ncol = 3)
```



```
ggsave(filename = "~/Dropbox/DimitriadisPohleWermuth/Binary Correlation/replication_BCor/application/pl
    height = 200 - adj_height, width = (1 + sqrt(5))/2 * 200 -
        adj_width, device = "pdf", units = "mm")

plot_contour_trivial(-0.7, c(0.01, 0.99), c(0.01, 0.99), 10,
    "P(A)", "P(B)", 40) + coord_fixed() + theme_minimal(base_size = base_size) +
    theme(axis.text.x = element_text(color = "black"), axis.text.y = element_text(color = "black"),
        axis.ticks = element_line(color = "black")) + plot_contour_binary(YuleQ,
    Cole.inv, -0.7, c(0.01, 0.99), c(0.01, 0.99), gridsize, "P(A)",
    "P(B)", 40) + coord_fixed() + theme_minimal(base_size = base_size) +
    theme(axis.text.x = element_text(color = "black"), axis.ticks.x = element_line(color = "black"),
        axis.text.y = element_blank(), axis.ticks.y = element_blank(),
        axis.title.y = element_blank()) + plot_contour_binary(Phi,
    Cole.inv, -0.7, c(0.01, 0.99), c(0.01, 0.99), gridsize, "P(A)",
    "P(B)", 40) + coord_fixed() + theme_minimal(base_size = base_size) +
    theme(axis.text.x = element_text(color = "black"), axis.ticks.x = element_line(color = "black"),
        axis.text.y = element_blank(), axis.ticks.y = element_blank(),
        axis.title.y = element_blank()) + plot_layout(guides = "collect",
    ncol = 3)
```



```
ggsave(filename = "~/Dropbox/DimitriadisPohleWermuth/Binary Correlation/replication_BCor/application/pl
    height = 200 - adj_height, width = (1 + sqrt(5))/2 * 200 -
        adj_width, device = "pdf", units = "mm")
```