Analysing the Expected Utility at Each Iteration

This script analyses our expected utility results in greater detail.

Setting everything up

```r
setwd("~/github/strategicvoting")
library(tidyverse)
library(rio)
library(gtools)
# library(devtools)

# Load data
load("output/files/1/85_winners_tbl.Rdata")
load("output/files/1/85_winners.Rdata")
load("output/big_list_2.RData")
vap <- read.csv("data/case_vap.csv", sep = "") # voting age pop.

# Load scripts and templates
source("code/utils/sv_theme_template.R")
source("code/utils/new_sv_iter.R")
source("code/prep_cses.R") # data prep
```

```
## big_list_na_omit prepared.
```

```r
# Merge winners and EU together
ddf <- winners_df %>%
    left_join(win_df)

# Set parameters
lambda <- 1
s <- 85
```

Looking At Expected Utility

First, let's print the expected utility for each case, and every iteration.

```r
# Calculate aggregates
eu_agg <- ddf %>%
    left_join(case_weight_tbl, by = c("case" = "case")) %>%
    group_by(iter, system) %>%
    summarise(value = weighted.mean(value, case_weight)) %>%
    mutate(iter = as.numeric(iter))
```

```
## `summarise()` has grouped output by 'iter'. You can override using the `.groups` argument.
```

```r
p1 <- ggplot(ddf, aes(iter, value)) +
    geom_line(aes(group = case), alpha = 0.05) +
    geom_line(
        data = eu_agg,
        alpha = 1,
        lwd = 1.1
    ) +
    scale_color_manual(values = c("#CC6600", "#004C99")) +
    facet_wrap(~system) +
    theme_tn()
```

p1
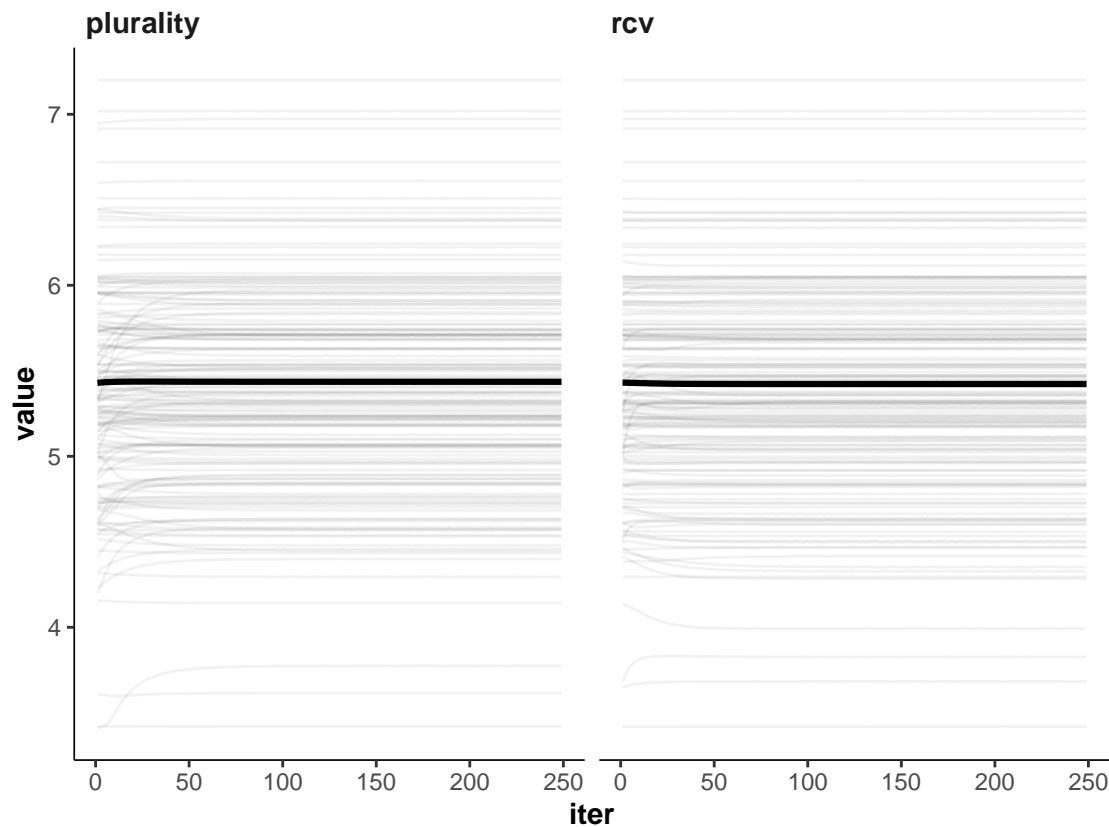
We observe that while the cross-case averages remain stable under both simulated systems, there are different kinds of movement under both: some cases increase in EU under plurality, others decrease in EU under plurality. The same is true for cases under RCV.

**Individual Cases' Expected Utilities Across Iterations**

Next, let's get a picture of how these cases behave at the 1st, and 60th iteration, respectively. The next chunk plots cases' EU at the 1st iteration, and the 60th iteration. (Filled) Points above the dotted line are those where the EU increases with iterations, while (hollow) points below the dotted line are those where EU decreases as the algorithm moves on.

```r
# Create scatterplots for EU
eu_scatter <- ddf %>%
    select(-V1, -V2, -V3) %>%
    filter(iter %in% c(1, 60)) %>%
    pivot_wider(
        values_from = value,
        names_from = iter,
        names_prefix = "iter"
    ) %>%
    mutate(decrease = iter60 < iter1)

p2 <- ggplot(eu_scatter, aes(iter1, iter60)) +
    geom_abline(slope = 1, intercept = 0, lty = "dotted") +
    geom_point(aes(color = system, shape = decrease), alpha = 0.6) +
    facet_wrap(. ~ system) +
```

```r
    scale_shape_manual(values = c(16, 21)) +
    labs(x = "Avg EU at 1st Iteration", y = "Avg EU at 60th Iteration") +
    theme_tn()
p2
```


plurality   **rcv**

system ● plurality ● rcv **decrease** ● FALSE ○ TRUE
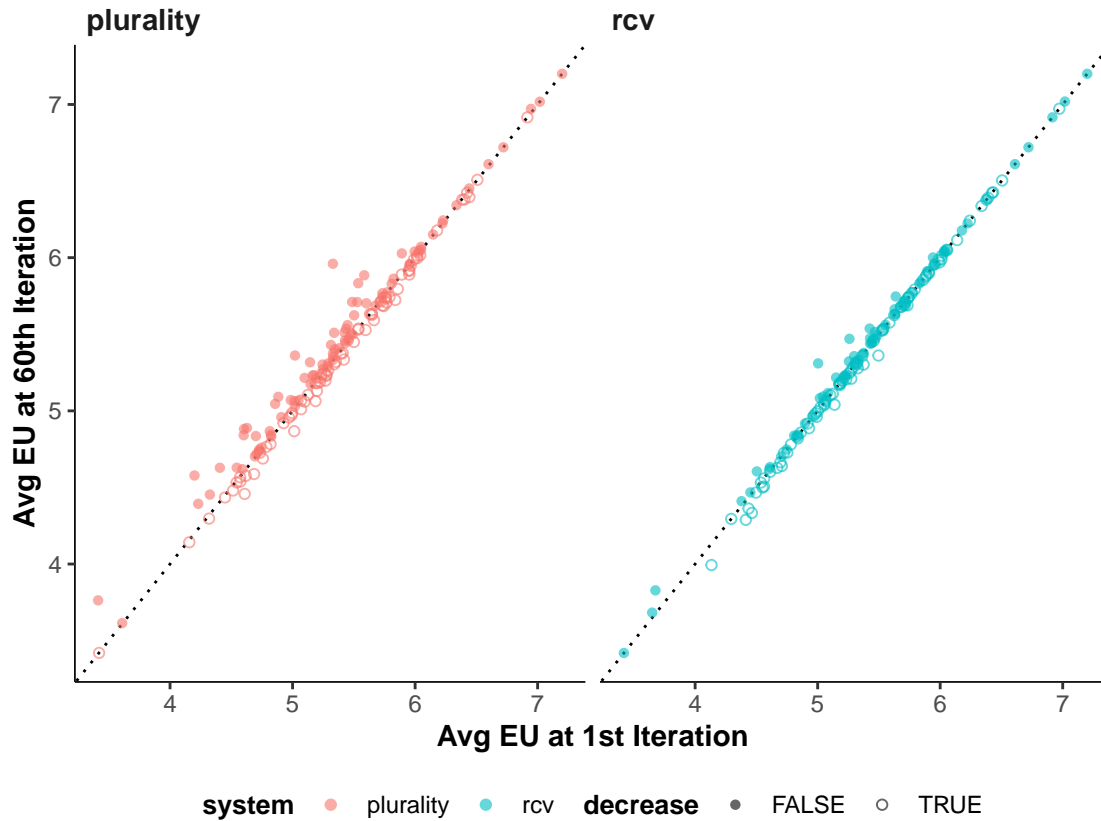
What this tells us (roughly) is that

- the magnitude of changes is seemingly greater under Plurality;
- most of the 'large' changes are on an upward trend.

NB. I also inspected a similar plot with RCV vs Plurality on the axes, faceted by iteration. That didn't give any more insight.

Changes in EU and Win Probability Divergence

Next, let's see how whether cases that have high divergence in expected utility between plurality and RCV are also cases that lead to different win probabilities.

For this, I calculate a distance metric between the win probability vectors.

```r
# For each case, compute difference in win probability by candidate
# Then calculate RMSE
diff_df <- ddf %>%
    group_by(case, iter) %>%
    summarise(
        diff_V1 = V1[system == "plurality"] - V1[system == "rcv"],
        diff_V2 = V2[system == "plurality"] - V2[system == "rcv"],
        diff_V3 = V3[system == "plurality"] - V3[system == "rcv"],
        diff_eu = value[system == "plurality"] - value[system == "rcv"]
```

```
    ) %>%
    mutate(vec_diff = sqrt(diff_V1^2 + diff_V2^2 + diff_V3^2))
```

`summarise()` has grouped output by 'case'. You can override using the `.groups` argument.

```
diff_df_pruned <- diff_df %>%
    filter(iter == 1 | iter == 60)
```

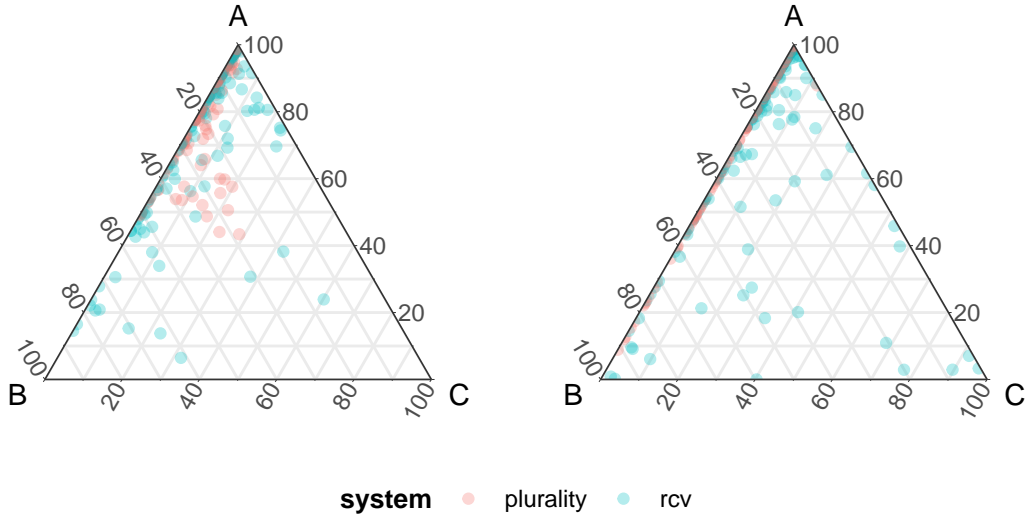Let's visualise the distribution of win probabilities in the ternary.

```
library(ggtern)
```

Registered S3 methods overwritten by 'ggtern':
method   from
grid.draw.ggplot ggplot2
plot.ggplot   ggplot2
print.ggplot   ggplot2

--
Remember to cite, run citation(package = 'ggtern') for further info.
--

##
Attaching package: 'ggtern'

The following objects are masked from 'package:ggplot2':
##
aes, annotate, ggplot, ggplotGrob, ggplot_build, ggplot_gtable,
ggsave, layer_data, theme_bw, theme_classic, theme_dark,
theme_gray, theme_light, theme_linedraw, theme_minimal, theme_void

```
winners_df_clean <- ddf %>%
    filter(iter == 1 | iter == 60) %>%
    rename(A = V1, B = V2, C = V3)

p3 <- ggtern(winners_df_clean, aes(B, A, C)) +
    # geom_line(aes(group = case), alpha = 0.2) +
    geom_point(aes(colour = system), alpha = 0.3) +
    facet_wrap(. ~ iter) +
    theme_tn()
p3
```

1   **60**

A   A
100   100

20   80   20   80

40   60   40   60

60   60
              40   40

80   80
                20   20

100   100
B   20   40   60   80   100 C   B   20   40   60   80   100 C

system ● plurality ● rcv

It's a little bit hard to see, but (perhaps unsurprisingly) the majority of cases are clustered towards the A vertex of the ternary. When inspecting things at the 1st iteration, there are both Plurality and RCV cases in the centre of the simplex. Moving to the 60th iteration, all Plurality cases moved towards the A/B edge of the simplex, while a minority of RCV cases still remain in the centre (or even shifted towards the C vertex!).

So, we should expect that cases that show a larger difference between their win probability vector under plurality and their win probability vector under RCV should also exhibit a greater difference in expected utility.

To calculate the difference in expected utility, I use the following convention:

$$\Delta_i = EU_i^{plur} - EU_i^{rcv}$$

where *EU* represents the (weighted) average of expected utility among voters in a case's given iteration. So positive values indicate that EU is higher in Plurality, while negative values indicate that EU is higher in RCV.

```
p3 <- ggplot(diff_df_pruned, aes(vec_diff, diff_eu)) +
    geom_abline(intercept = 0, slope = 0, lty = "dotted") +
    geom_line(aes(group = case), alpha = 0.1) +
    geom_point(aes(colour = as.factor(iter)), alpha = 0.7) +
    labs(
        x = "RMSE Win Probability",
        y = "Delta EU",
        colour = "Iteration"
    ) +
    theme_tn()
p3
```
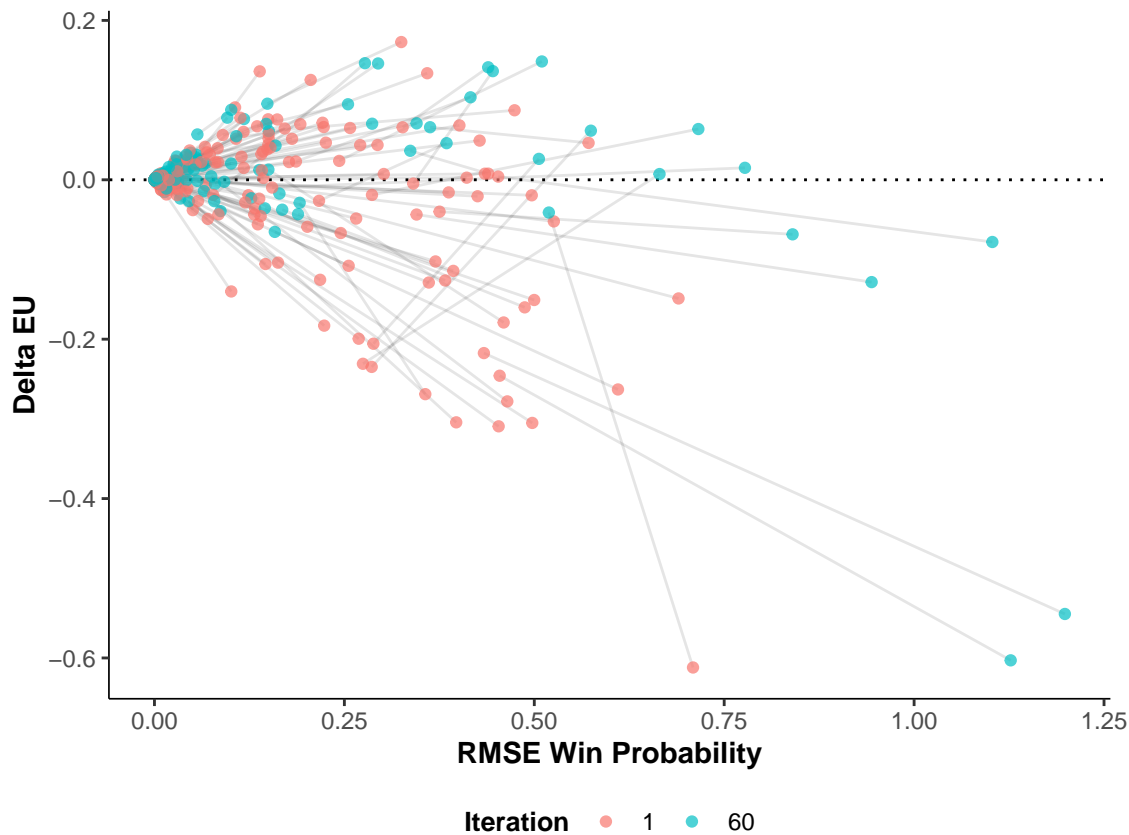
The thin lines connect the data points for the same case. In the first iteration, for most cases where RCV and Plurality lead to different win probabilities, the average expected utility is higher under RCV than it is under Plurality. When looking at the 60th iteration, however, the picture changes: bar a few extreme outliers, most cases where the electoral systems diverge bring higher average expected utility under Plurality.

**Visualising Changes When Voters Become More Strategic**

First, we Δ_1 on the x-axis, and $\Delta_6 0$ on the y-axis. We notice that there are observations both in the upper right quadrant (plurality yields higher AEU no matter how strategic voters are) as well as in the lower left quadrant (RCV yields higher AEU no matter how strategic voters are).

```r
# Make delta data wide
diff_df_pruned_wide <- diff_df_pruned %>%
    pivot_wider(values_from = diff_eu, id_cols = c(case), names_from = iter, names_prefix = "delta_eu_")

px1 <- ggplot(diff_df_pruned_wide, aes(delta_eu_1, delta_eu_60)) +
    geom_point(colour = "#727272") +
    geom_hline(yintercept = 0, lty = "dotted") +
    geom_vline(xintercept = 0, lty = "dotted") +
    geom_smooth(method = "lm") +
    geom_abline(slope = 1, intercept = 0, lty = "dashed") +
    labs(
        x = "Delta EU (1st)",
        y = "Delta EU (60th)"
    ) +
    annotate(x = -0.2, y = -0.3, geom = "text", label = "RCV always higher AEU") +
    annotate(x = 0.15, y = 0.05, geom = "text", label = "Plur always \n higher AEU") +
    theme_tn()
```

```
## `geom_smooth()` using formula 'y ~ x'
```
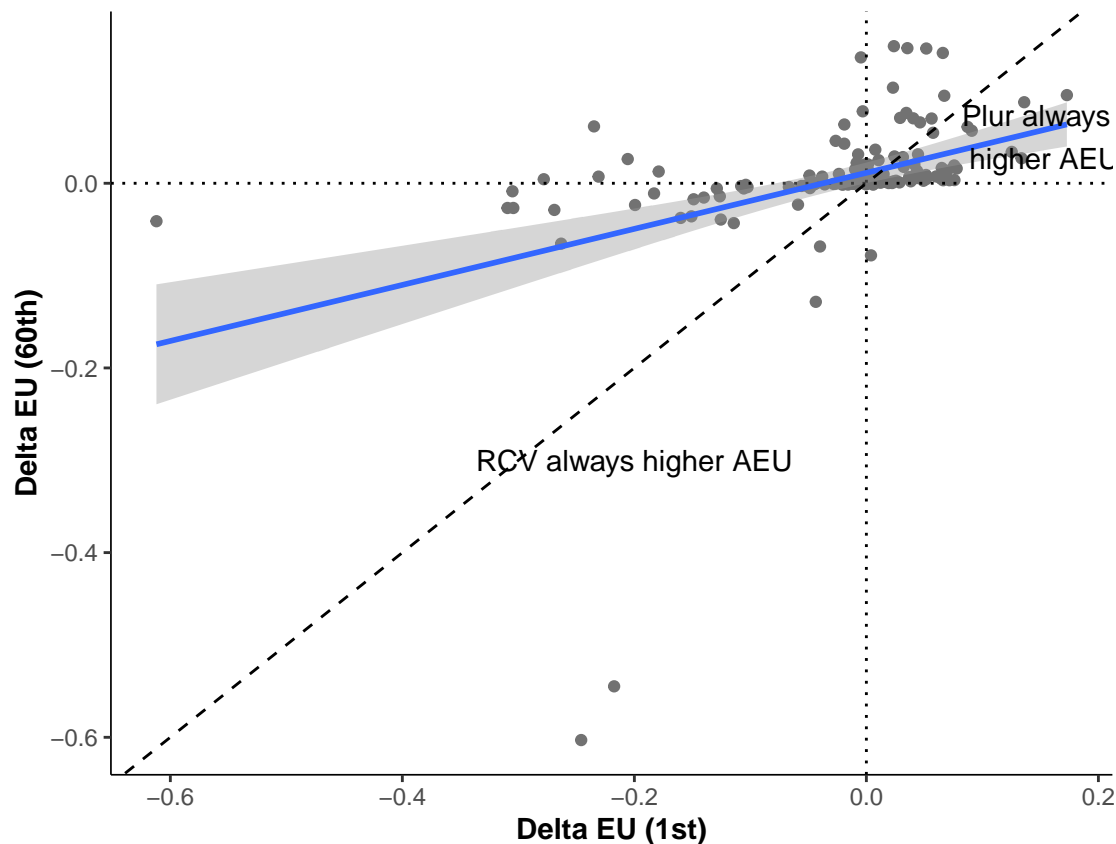

This plot tells us that when voters behave strategically, the average expected utility difference between systems is attenuated.

More to the point, what I think might be the most interesting comparison is comparing AEU under plurality to AEU under RCV with naive voters. This is a relevant comparison because we'd probably think that voters under RCV are far less likely to be strategic (since it has a higher cost in terms of information and computation).

When we compare the AEU difference between Plurality and RCV at the first iteration, and the AEU difference between Plurality at the 60th and RCV at the 1st iteration, we see that, on average, if voters are strategic under Plurality (but not under RCV), being strategic under Plurality almost fully attenuates the difference between the two electoral systems.

```r
# Compare plurality strategy to RCV "baseline"
diff_df_extra <- ddf %>%
    group_by(case) %>%
    summarise(
        diff_p1_rcv1 = value[system == "plurality" & iter == 1] - value[system == "rcv" & iter == 1],
        diff_p60_rcv1 = value[system == "plurality" & iter == 60] - value[system == "rcv" & iter == 1],
        diff_p60_rcv60 = value[system == "plurality" & iter == 60] - value[system == "rcv" & iter == 60]
        diff_p60_p1 = value[system == "plurality" & iter == 60] - value[system == "plurality" & iter ==
        diff_rcv60_rcv1 = value[system == "rcv" & iter == 60] - value[system == "rcv" & iter == 1]
    ) %>%
    left_join(
        case_weight_tbl
```

```
    )

## Joining, by = "case"
library(questionr)
avg_p1_rcv1 <- wtd.mean(diff_df_extra$diff_p1_rcv1)
avg_p60_rcv60 <- wtd.mean(diff_df_extra$diff_p60_rcv60)
avg_p60_p1 <- wtd.mean(diff_df_extra$diff_p60_p1)
avg_rcv60_rcv1 <- wtd.mean(diff_df_extra$diff_rcv60_rcv1)

# Make extra comparison wide
diff_df_pruned_wide <- diff_df_pruned %>%
    pivot_wider(values_from = diff_eu, id_cols = c(case), names_from = iter, names_prefix = "delta_eu_")



px2 <- ggplot(diff_df_extra, aes(diff_p1_rcv1, diff_p60_rcv1)) +
    geom_point() +
    geom_hline(yintercept = 0, lty = "dotted") +
    geom_vline(xintercept = 0, lty = "dotted") +
    geom_smooth(method = "lm") +
    geom_abline(slope = 1, intercept = 0) +
    labs(
        x = "Delta EU (P 1st, RCV 1st)",
        y = "Delta EU (P 60th, RCV 1st)"
    ) +
    annotate(x = -0.2, y = -0.3, geom = "text", label = "RCV always higher AEU") +
    annotate(x = 0.15, y = 0.05, geom = "text", label = "Plur always \n higher AEU") +
    theme_tn()
# px2

px3 <- ggplot(diff_df_extra, aes(diff_p1_rcv1, diff_p60_rcv60)) +
    geom_point(colour = "#727272") +
    geom_hline(yintercept = 0, lty = "dotted") +
    geom_hline(yintercept = avg_p60_rcv60) +
    geom_vline(xintercept = 0, lty = "dotted") +
    geom_vline(xintercept = avg_p1_rcv1) +
    geom_smooth(method = "lm", se = FALSE) +
    geom_abline(slope = 1, intercept = 0, lty = "dotted") +
    labs(
        x = "Delta EU (P 1st, RCV 1st)",
        y = "Delta EU (P 60th, RCV 60st)"
    ) +
    theme_tn() +
    ggtitle("Across Systems")
# px3

px4 <- ggplot(diff_df_extra, aes(diff_p60_p1, diff_rcv60_rcv1)) +
    geom_point(colour = "#727272") +
    geom_hline(yintercept = 0, lty = "dotted") +
    geom_hline(yintercept = avg_rcv60_rcv1) +
    geom_vline(xintercept = 0, lty = "dotted") +
    geom_vline(xintercept = avg_p60_p1) +
    geom_smooth(method = "lm", se = FALSE) +
```

```
    geom_abline(slope = 1, intercept = 0, lty = "dashed") +
    labs(
        x = "Delta EU (P 60st, P 1st)",
        y = "Delta EU (RCV 60th, RCV 1th)"
    ) +
    theme_tn() +
    ggtitle("Across Iterations")
# px4

library(gridExtra)
```
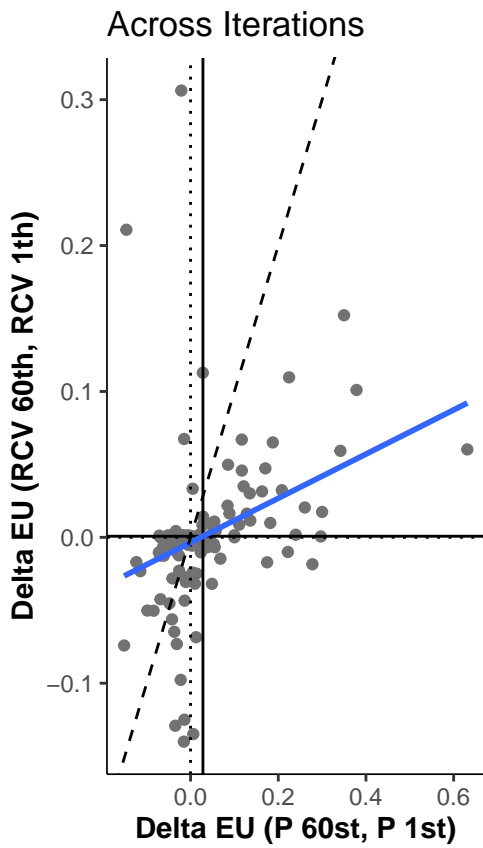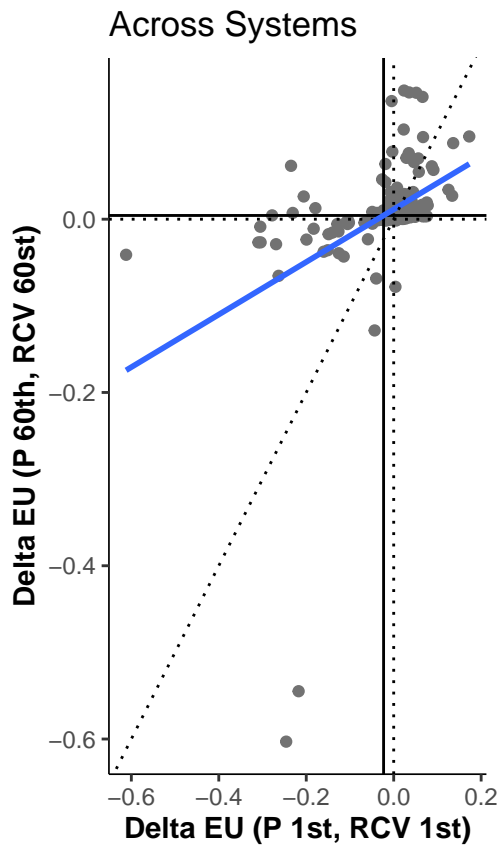
```
##
## Attaching package: 'gridExtra'

## The following objects are masked from 'package:ggtern':
##
##      arrangeGrob, grid.arrange

## The following object is masked from 'package:dplyr':
##
##      combine
```

```
px_comb <- gridExtra::grid.arrange(px3, px4, ncol = 2)
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
px_comb
```

```
## TableGrob (1 x 2) "arrange": 2 grobs
##   z     cells    name            grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
```

Finally, we produce the following figure:

Inspecting Extreme Cases in Greater Detail

Finally, let's look at some of the most extreme cases (in terms of win probability divergence), which are really nice examples of divided majority vs single-peaked preference profiles. In the previous figure, we see three cases in the bottom right corner of the plot: one observation (Romania) from the 1st iteration that subsequently moves towards the centre again, and two observations (both Switzerland – focussing on CHE_1999 henceforth) that start much closer to the centre in the 1st observation before diverging strongly in terms of win probabilities. (I also look at GBR_2005 further below)

```
p4 <- ggplot(
    diff_df_pruned %>%
        filter(case %in% c("CHE_1999", "ROU_2004", "GBR_2005")),
    aes(vec_diff, diff_eu)
) +
    geom_abline(intercept = 0, slope = 0, lty = "dotted") +
    geom_line(aes(group = case), alpha = 0.1) +
    geom_point(aes(colour = as.factor(iter)), alpha = 0.7) +
    geom_text(aes(label = case)) +
    labs(
        x = "RMSE Win Probability",
        y = "Delta EU",
        colour = "Iteration"
    ) +
    theme_tn()
p4
```
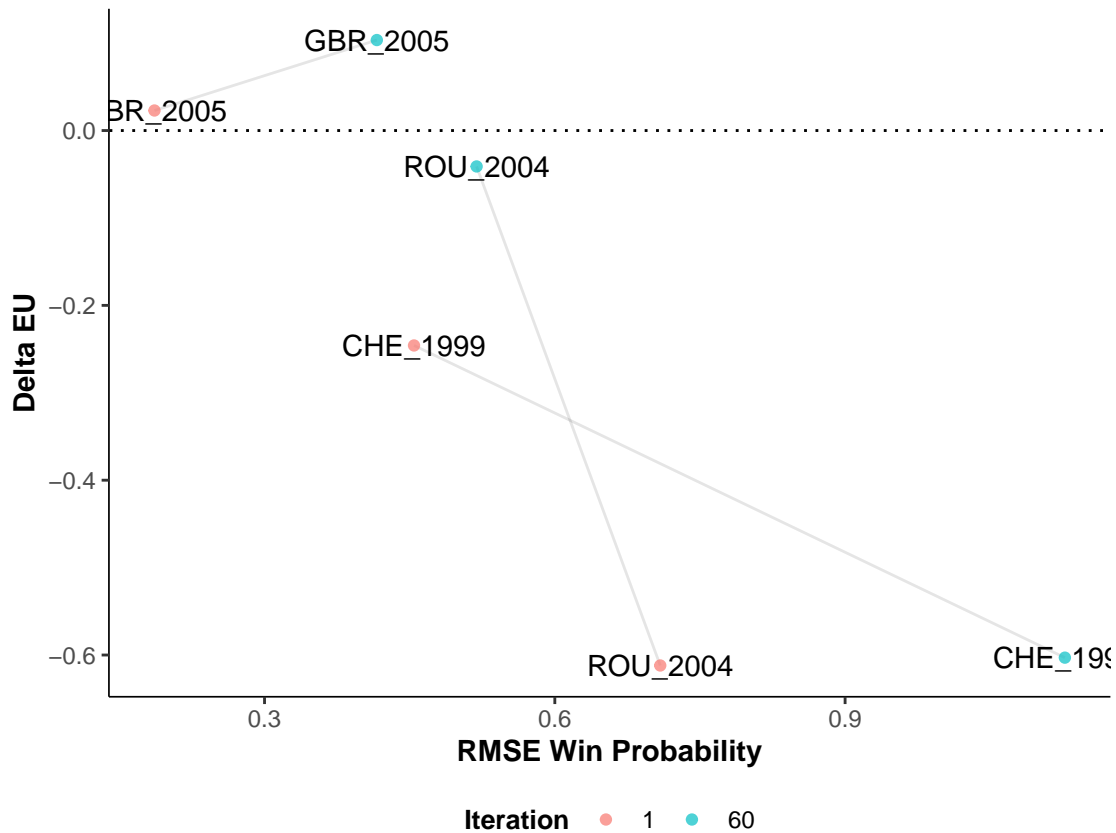
Because the two cases also move in opposite directions as we move through the iterations, they are also nice illustrations of cases where the expected utility difference between systems either converges towards 0 or diverges towards an extreme.

What is going on?

```
winners_df_clean %>%
    filter(case %in% c("CHE_1999", "ROU_2004")) %>%
    arrange(case, iter, system)
```

```
##        case       A       B       C iter    system    value
## 1 CHE_1999 0.44063 0.32803 0.23134    1 plurality 5.013498
## 2 CHE_1999 0.23921 0.15884 0.60195    1       rcv 5.259350
## 3 CHE_1999 0.58559 0.41441 0.00000   60 plurality 4.867177
## 4 CHE_1999 0.07053 0.01143 0.91804   60       rcv 5.470190
## 5 ROU_2004 0.64042 0.27631 0.08327    1 plurality 5.328932
## 6 ROU_2004 0.06459 0.61562 0.31979    1       rcv 5.940896
## 7 ROU_2004 0.08871 0.91129 0.00000   60 plurality 5.960088
## 8 ROU_2004 0.00012 0.59657 0.40331   60       rcv 6.001161
```

```
big_list_na_omit$CHE_1999$v_vec
```

```
## [1] 0.09080525 0.25642490 0.10108509 0.23186750 0.16447744 0.15533981
```

```
big_list_na_omit$ROU_2004$v_vec
```

```
## [1] 0.18746120 0.19180633 0.05710739 0.26877716 0.04345127 0.25139665
```

In the **CHE__1999** case, both A and B voters strongly prefer C to each other's party, while C voters are evenly split (so classic single-peaked case with C being the centrist candidate). Under plurality, C voters abandon their favourite party in equal parts towards A and B, leading to both A and B having a decent shot

11

at winning (and leaving supporters of the respective other party dissatisfied). Under RCV, on the other hand, the algorithm converges upon C as the clear consensus winner, which increases average expected utility.

In the **ROU_2004** case, A voters are equally split in B vs C as second preference. B voters strongly prefer C as second preference, and C voters strongly prefer B voters. So this is a classic case of divided majority. Under plurality, C voters move towards voting for B, thus making them the clear winner in later iterations, and improving EU (as A, which is disliked by both B and C is rendered unlikely to win). Under RCV, on the other hand, both B and C remain competitive.

To extend the picture further, let's look at a case that becomes more favourable to Plurality as iterations progress.

```r
winners_df_clean %>%
    filter(case == "GBR_2005") %>%
    arrange(case, iter)
```

```
##        case       A       B       C iter    system    value
## 1 GBR_2005 0.80547 0.02137 0.17316    1       rcv 4.992581
## 2 GBR_2005 0.91779 0.05398 0.02823    1 plurality 5.015501
## 3 GBR_2005 0.69495 0.00308 0.30197   60       rcv 4.960689
## 4 GBR_2005 0.98073 0.01927 0.00000   60 plurality 5.064211
```

```r
winners_df_clean %>%
    filter(case == "CAN_2011") %>%
    arrange(case, iter)
```

```
##        case       A       B       C iter    system    value
## 1 CAN_2011 0.56267 0.34217 0.09516    1       rcv 5.137734
## 2 CAN_2011 0.82384 0.16465 0.01151    1 plurality 5.203644
## 3 CAN_2011 0.27456 0.47149 0.25395   60       rcv 5.040024
## 4 CAN_2011 0.62093 0.37907 0.00000   60 plurality 5.181194
```

```r
big_list_na_omit$GBR_2005$v_vec
```

```
## [1] 0.11988717 0.31170663 0.10155148 0.19040903 0.18053597 0.09590973
```

Both A and B voters mostly prefer C to each other; C voters prefer A. (single-peaked case). Under Plurality, A consolidates its win probability and becomes the undisputed winner. The expected utility increases slightly, but it's hard to tell whether that is because of the decrease in Pr(B win) or due to other factors. Under RCV, C's chance of winning increases in later iterations. Even though GBR_2005 is a single-peaked profile, this does not translate into an increase in expected utility. This could be, for example, because there is a segment of BAC voters that just really hates C.

I'm not 100% sure if this case is fully representative of the family of cases that move towards the upper right in the plot.