114th Roll Call Data

```r
rm(list = ls())

sapply(c("knitr", "buds", "coda", "dplyr", "ggplot2", "MCMCglmm",
         "rstan", "viridis"), require, character.only = TRUE)
```

```
## Loading required package: knitr

## Loading required package: buds

## Loading required package: Rcpp

## Loading required package: coda

## Loading required package: dplyr

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

## Loading required package: ggplot2

## Loading required package: MCMCglmm

## Loading required package: Matrix

## Loading required package: ape

## Loading required package: rstan

## Loading required package: StanHeaders

## rstan (Version 2.15.1, packaged: 2017-04-19 05:03:57 UTC, GitRev: 2e1f913d3ca3)

## For execution on a local, multicore CPU with excess RAM we recommend calling
## rstan_options(auto_write = TRUE)
## options(mc.cores = parallel::detectCores())

##
## Attaching package: 'rstan'

## The following object is masked from 'package:coda':
##
##     traceplot

## Loading required package: viridis

## Loading required package: viridisLite

##    knitr     buds     coda    dplyr  ggplot2 MCMCglmm    rstan  viridis
##     TRUE     TRUE     TRUE     TRUE     TRUE     TRUE     TRUE     TRUE
```

```r
# Save generated figures
opts_chunk$set(fig.path = paste0("./roll_call/"), dev='png')
# Functions
```

```r
source("../R/distcomps.R")
source("../R/get_data_to_plot.R")
source("../R/plot_utils.R")
```

```r
# Options
rstan_options(auto_write = TRUE)
options(mc.cores = parallel::detectCores())
theme_set(theme_classic())
theme_update(text=element_text(size=20))

# Parameters
min_row_sum <- 100
min_row_prevalence <- 5
B <- 100
min_sigma <- 0.05
hparams <- list(
  "gamma_tau"= 2.5,
  "gamma_epsilon" = 2.5,
  "gamma_bias" = 2.5,
  "gamma_rho" = 2.5,
  "min_sigma" = min_sigma
)
nfeatures <- 500
K <- 10
```

Roll Call

```r
# Load data files
countTable_default_file <- "../data/114_US_Senate_binVotes.csv"
sampleData_default_file <- "../data/114_US_Senate_legisData.csv"

sampleData <- read.csv(sampleData_default_file, row.names = 1)
covariate_name <- "party"
sample_covariate <- sampleData[, covariate_name]

X <- read.csv(countTable_default_file, row.names = 1)
D0 <- generic_dist(X, method = "exp manhattan", log_trans = FALSE)
D <- D0
```

```r
set.seed(1)
buds_seed <- sample.int(.Machine$integer.max, 1)
budsFit <- buds::fit_buds(D, K = K, method = "vb", hyperparams = hparams,
                          init_from = "random", seed = buds_seed,
                          tol_rel_obj = 0.005)
```

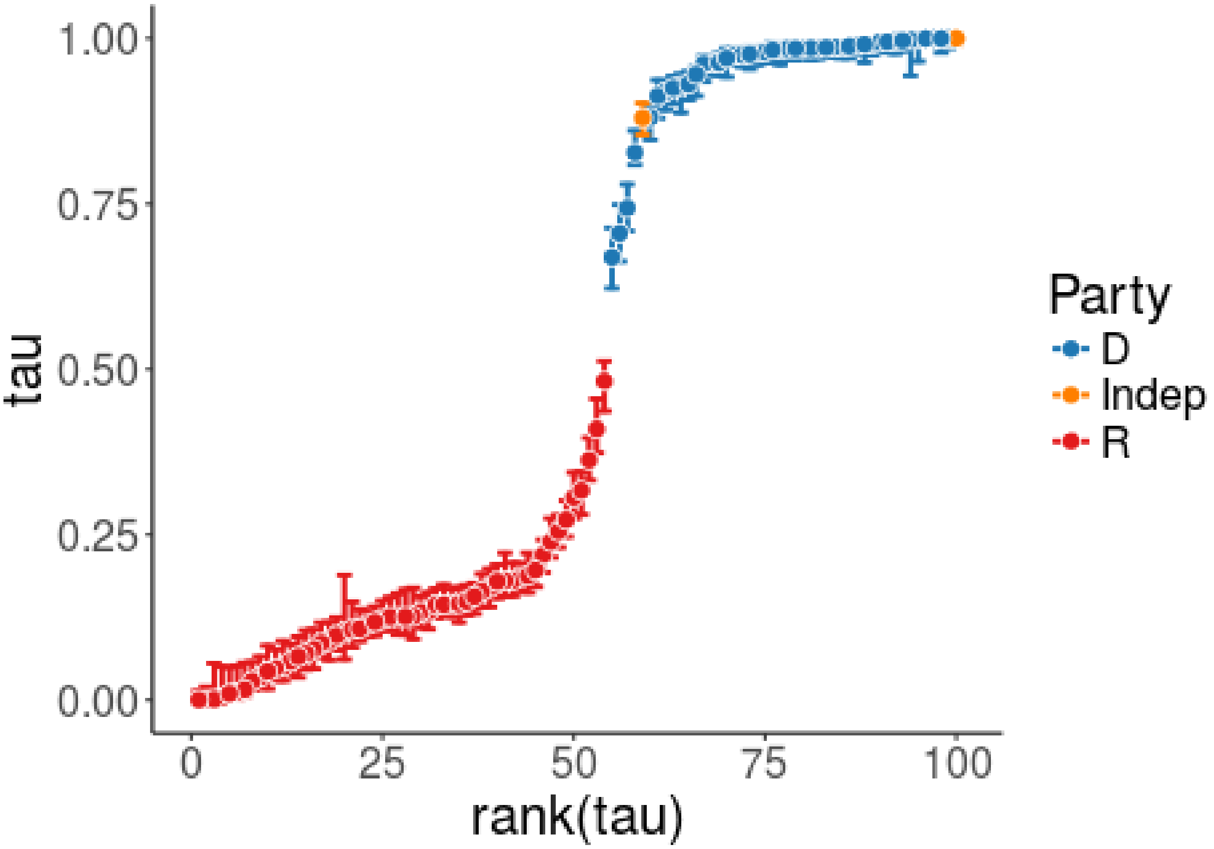
```
## ------------------------------------------------------------
## EXPERIMENTAL ALGORITHM:
##   This procedure has not been thoroughly tested and may be unstable
##   or buggy. The interface is subject to change.
## ------------------------------------------------------------
##
##
```

```
##
## Gradient evaluation took 0.002988 seconds
## 1000 transitions using 10 leapfrog steps per transition would take 29.88 seconds.
## Adjust your expectations accordingly!
##
##
## Begin eta adaptation.
## Iteration:   1 / 250 [  0%]  (Adaptation)
## Iteration:  50 / 250 [ 20%]  (Adaptation)
## Iteration: 100 / 250 [ 40%]  (Adaptation)
## Success! Found best value [eta = 100] earlier than expected.
##
## Begin stochastic gradient ascent.
##   iter       ELBO   delta_ELBO_mean   delta_ELBO_med   notes
## ------------------------------------------------------------
## EXPERIMENTAL ALGORITHM:
##   This procedure has not been thoroughly tested and may be unstable
##   or buggy. The interface is subject to change.
## ------------------------------------------------------------
##
##
##
## Gradient evaluation took 0.003331 seconds
## 1000 transitions using 10 leapfrog steps per transition would take 33.31 seconds.
## Adjust your expectations accordingly!
##
##
## Begin eta adaptation.
## Iteration:   1 / 250 [  0%]  (Adaptation)
## Iteration:  50 / 250 [ 20%]  (Adaptation)
## Iteration: 100 / 250 [ 40%]  (Adaptation)
## Iteration: 150 / 250 [ 60%]  (Adaptation)
## Iteration: 200 / 250 [ 80%]  (Adaptation)
## Success! Found best value [eta = 1] earlier than expected.
##
## Begin stochastic gradient ascent.
##   iter       ELBO   delta_ELBO_mean   delta_ELBO_med   notes
##    100     -2e+04             1.000            1.000
##    200     -6e+02            16.736           32.473
##    300      6e+03            11.520            1.086
##    400      8e+03             8.696            1.086
##    500      8e+03             6.961            1.000
##    600      9e+03             5.809            1.000
##    700      9e+03             4.979            0.224
##    800      9e+03             4.357            0.224
##    900      9e+03             3.874            0.045
##   1000      9e+03             3.487            0.045
##   1100      9e+03             3.387            0.023   MAY BE DIVERGING... INSPECT ELBO
##   1200      9e+03             0.140            0.006
##   1300      9e+03             0.031            0.005
##   1400      9e+03             0.009            0.004   MEDIAN ELBO CONVERGED
##
## Drawing a sample of size 1000 from the approximate posterior...
## COMPLETED.
```

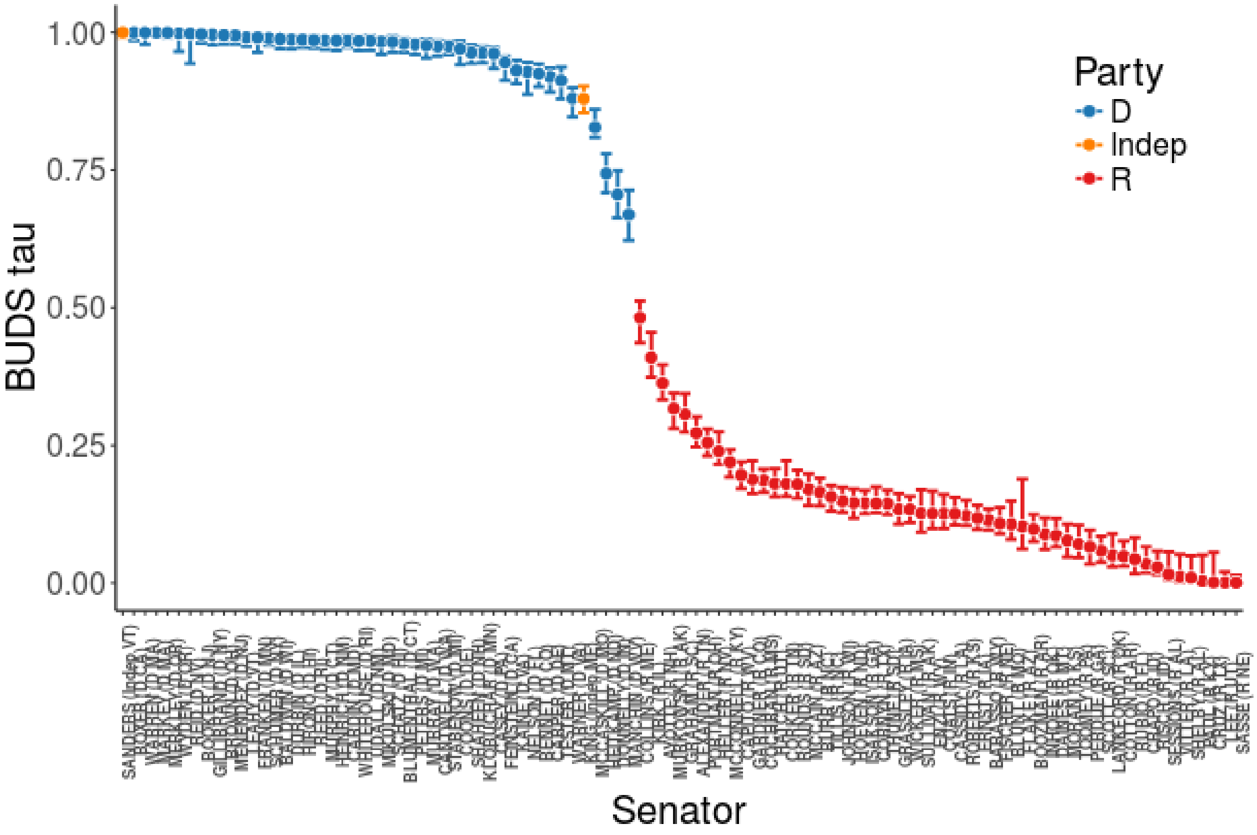
```r
budsParams <- (rstan::extract(budsFit$fit_buds))
tau_df <- get_tau_df(budsParams, prob = 0.95)
```

```r
party_cols <- c("D" = "#1f78b4", "R" = "#e31a1c", "Indep" = "#ff7f00")
plt <- plot_buds_1D(tau_df, covariate = NULL,
                    color = sample_covariate,
                    color_label = covariate_name,
                    idxBigger = NULL)
plt + geom_errorbar(aes(ymin = tau_lower, ymax = tau_upper), lwd = 0.9, width = 2) +
  geom_point(aes(fill = color), color = "white", pch = 21, size = 3) +
  scale_color_manual(name = "Party", values = party_cols) +
  scale_fill_manual(name = "Party", values = party_cols)
```

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

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



```r
sampleData$Legis <- rownames(sampleData)
sampleData$Legis <- factor(sampleData$Legis,
                           levels = sampleData$Legis[order(-tau_df$tau)])
(ptau1 <- tau_df %>%
    ggplot(aes(x = sampleData$Legis, y = tau, col = sampleData$party)) +
    geom_errorbar(aes(ymin = tau_lower, ymax = tau_upper), width = 1, lwd = 0.9) +
    geom_point(pch = 21, aes(fill = sampleData$party), color = "white", size = 3) +
    scale_fill_manual(name = "Party", values = party_cols) +
    scale_color_manual(name = "Party", values = party_cols) +
    ylab("BUDS tau") + xlab("Senator") +
    theme(text = element_text(size = 20), legend.position = c(0.9, 0.8),
          axis.text.x = element_text(angle = 90, size = 8, face = "bold")))
```

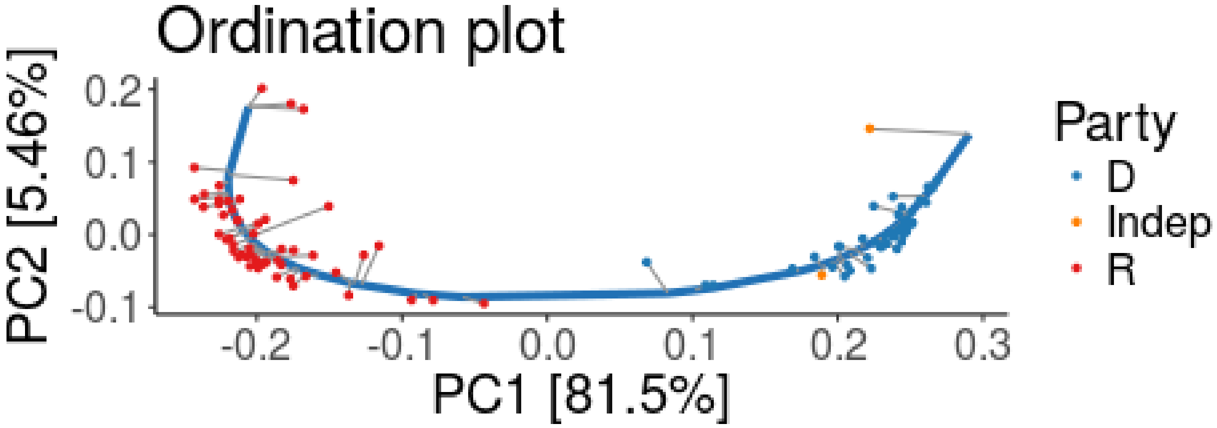


Bootstrap MDS ordering

```
# Plot Ord
plot_ord <- function(ord_res, ord_eig = NULL, size = 1,
                     colData = NULL, colLabel = "Variable",
                     title = "Ordination plot", prin_curve = FALSE,
                     edgesCol = "grey57", pathCol = "#2171B5",
                     lwd = 1.5, ...) {
  if (!is.null(ord_eig)) {
    ord_eig <- 100 * ord_eig /sum(ord_eig)
    ord_eig <- signif(ord_eig, digits = 3)
  }
  X <- data.frame(ord_res)
  colnames(X) <- paste0("X", 1:ncol(X))
  p <- ggplot(X, aes(X1, X2)) + ggtitle(title)

  if(prin_curve) {
    prin_curve <- princurve::principal.curve(as.matrix(X), plot = FALSE, ...)
    fittedLine <- data.frame(prin_curve$s[prin_curve$tag, ])
    p <- p + geom_path(data = fittedLine, col = pathCol, lwd = lwd) +
      geom_segment(aes(xend = prin_curve$s[, 1], yend = prin_curve$s[, 2]),
                   size = 0.5, col = edgesCol)
  }
  if (!is.null(colData)) {
    p <- p + geom_point(aes(color = colData), size = size) +
      scale_color_viridis(name = colLabel, discrete = !is.numeric(colData))
  } else {
    p <- p + geom_point(size = size)
  }
  if (!is.null(ord_eig)){
    eig_ratio =  ord_eig[2]/ord_eig[1]
    p <- p + xlab(paste0("PC1 [", ord_eig[1], "%]")) +
      ylab(paste0("PC2 [", ord_eig[2], "%]")) +
      coord_fixed(ratio = max(0.5, eig_ratio))
  }
  return(list(plot = p, fit.prin_curve = prin_curve))
}
```

```
cmdsRes <- cmdscale(D, k = 10, eig = TRUE)
ord_eig <- signif(cmdsRes$eig/sum(cmdsRes$eig) *100, digits = 3)
```

```
ordRes <- plot_ord(data.frame(cmdsRes$points), ord_eig,
         colData = sampleData$party, prin_curve = T)
ordRes$plot + scale_color_manual(name = "Party", values = party_cols)
```



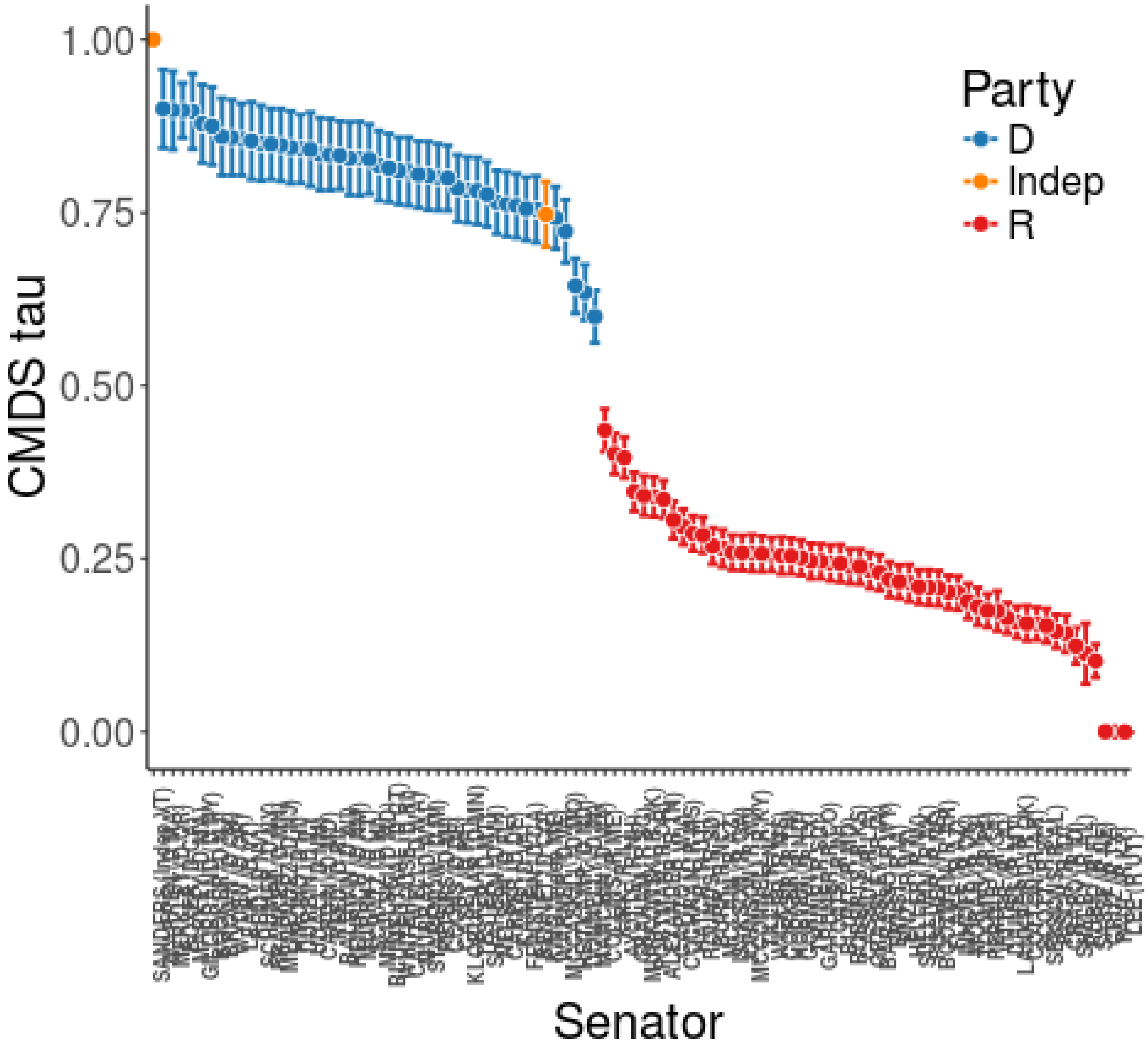
```
tau0 <- ordRes$fit.prin_curve$lambda
tau0 <- (tau0 - min(tau0))/diff(range(tau0))
```

```
set.seed(123)
B <- 500
nSamples <- floor(0.8*ncol(D))

tau.boots <- matrix(NA, nrow = B, ncol = ncol(D))

for(i in 1:B) {
  if (i %% 50 == 0) print(paste("Iteration:", i))
  idx <- sample(1:ncol(D), nSamples)
  icmds <- cmdscale(D[idx, idx], k = 10, eig = TRUE)
  prin_curve <- princurve::principal.curve(icmds$points, plot = FALSE)
  itau <- prin_curve$lambda
  itau <- (itau - min(itau))/diff(range(itau))
  R1 <- cor(itau, tau0[idx], method = "spearman")
  if (R1 < 0) itau <- 1 - itau
  tau.boots[i, idx] <- itau
}
```

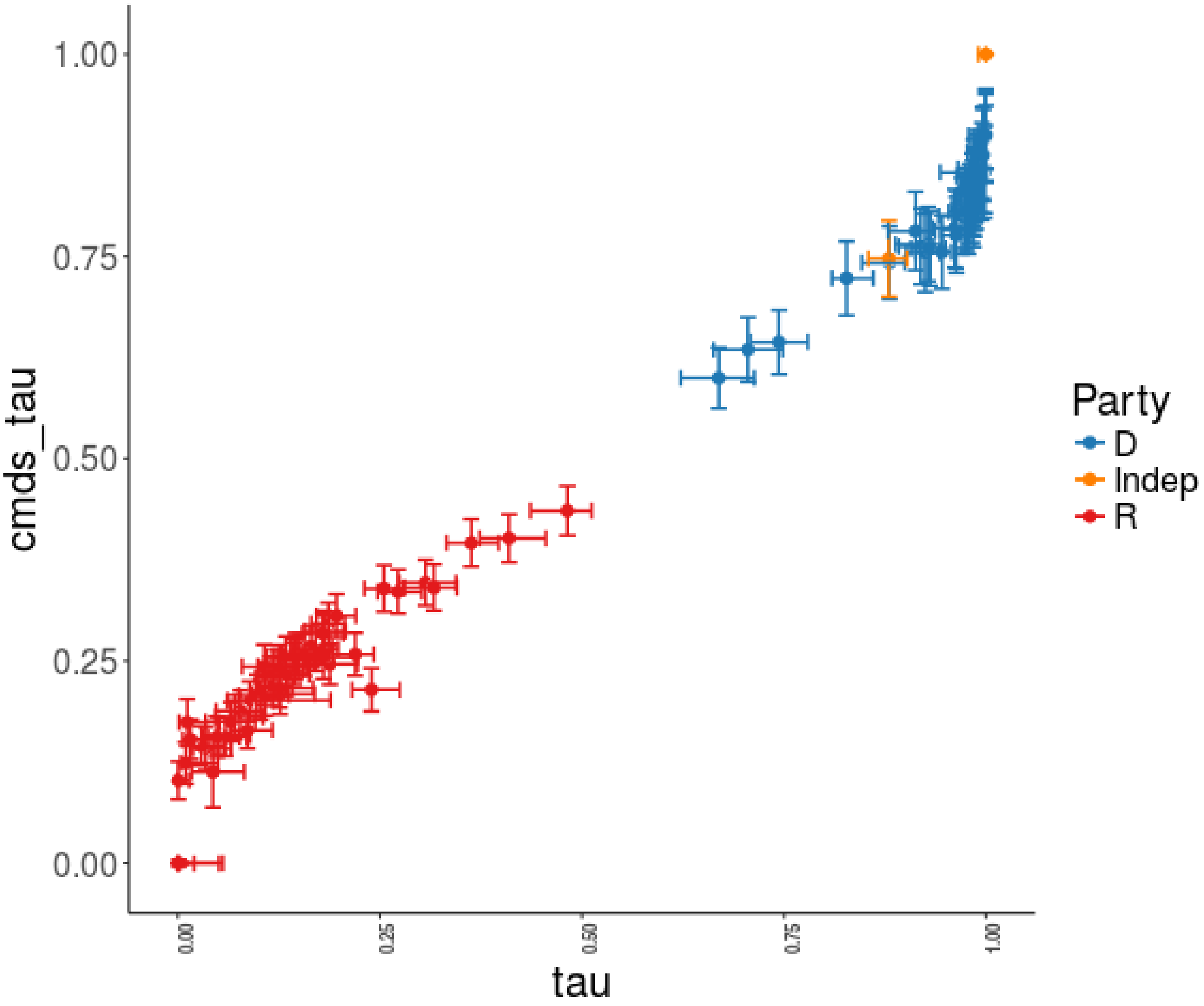
```
## [1] "Iteration: 50"
## [1] "Iteration: 100"
## [1] "Iteration: 150"
## [1] "Iteration: 200"
## [1] "Iteration: 250"
## [1] "Iteration: 300"
## [1] "Iteration: 350"
## [1] "Iteration: 400"
## [1] "Iteration: 450"
## [1] "Iteration: 500"
```

```
cmds_df <- data.frame(Senator = rownames(sampleData),
                       tau = colMeans(tau.boots, na.rm = TRUE))
sd_tau <- apply(tau.boots, 2, sd, na.rm = TRUE)
cmds_df$tau_upper <- cmds_df$tau + sd_tau
cmds_df$tau_lower <- cmds_df$tau - sd_tau
```

```
cmds_legis <- factor(sampleData$Legis,
                     levels = sampleData$Legis[order(-cmds_df$tau)])

(ptau_cmds <- cmds_df %>%
    ggplot(aes(x = cmds_legis, y = tau, col = sampleData$party)) +
    geom_errorbar(aes(ymin = tau_lower, ymax = tau_upper), width = 1, lwd = 0.9) +
    geom_point(pch = 21, aes(fill = sampleData$party), color = "white", size = 3) +
    scale_fill_manual(name = "Party", values = party_cols) +
    scale_color_manual(name = "Party", values = party_cols) +
    ylab("CMDS tau") + xlab("Senator") +
    theme(text = element_text(size = 20), legend.position = c(0.9, 0.8),
          axis.text.x = element_text(angle = 90, size = 8, face = "bold")))
```



```
colnames(cmds_df) <- c("senator", "cmds_tau", "cmds_tau_upper", "cmds_tau_lower")
df <- cbind(tau_df, cmds_df, party = sampleData$party)
rownames(df) <- NULL
(plt <- ggplot(df, aes(x = tau, y = cmds_tau, fill = party, color = party)) +
                 geom_point(color = "white", pch = 21, size = 3) +
                 scale_fill_manual(name = "Party", values = party_cols) +
                 scale_color_manual(name = "Party", values = party_cols) +
   geom_errorbar(aes(ymin = cmds_tau_lower, ymax = cmds_tau_upper), width = 0.02, lwd = 0.7) +
   geom_errorbarh(aes(xmin = tau_lower, xmax = tau_upper), height = 0.02, lwd = 0.7) +
   theme(text = element_text(size = 20),
         axis.text.x = element_text(angle = 90, size = 8, face = "bold"))
   + coord_fixed())
```



```r
ordered <- data.frame(buds = levels(sampleData$Legis),
                      cmds = levels(cmds_legis))
```

```r
kable(ordered)
```

buds	cmds
SANDERS (Indep VT)	SANDERS (Indep VT)
BOXER (D CA)	WARREN (D MA)
WARREN (D MA)	MERKLEY (D OR)
MARKEY (D MA)	BOXER (D CA)
MERKLEY (D OR)	MARKEY (D MA)
WYDEN (D OR)	GILLIBRAND (D NY)
REID (D NV)	BOOKER (D NJ)
BOOKER (D NJ)	BROWN (D OH)
GILLIBRAND (D NY)	WYDEN (D OR)
BROWN (D OH)	LEAHY (D VT)
MENENDEZ (D NJ)	REID (D NV)
LEAHY (D VT)	SCHUMER (D NY)
FRANKEN (D MN)	FRANKEN (D MN)
SCHUMER (D NY)	MENENDEZ (D NJ)
BALDWIN (D WI)	HIRONO (D HI)

buds	cmds
DURBIN (D IL)	DURBIN (D IL)
HIRONO (D HI)	UDALL (D NM)
REED (D RI)	CARDIN (D MD)
MURPHY (D CT)	REED (D RI)
HEINRICH (D NM)	BALDWIN (D WI)
CARDIN (D MD)	HEINRICH (D NM)
WHITEHOUSE (D RI)	MURPHY (D CT)
UDALL (D NM)	SCHATZ (D HI)
MIKULSKI (D MD)	MIKULSKI (D MD)
SCHATZ (D HI)	BLUMENTHAL (D CT)
BLUMENTHAL (D CT)	WHITEHOUSE (D RI)
PETERS (D MI)	CANTWELL (D WA)
MURRAY (D WA)	MURRAY (D WA)
CANTWELL (D WA)	STABENOW (D MI)
STABENOW (D MI)	PETERS (D MI)
COONS (D DE)	COONS (D DE)
SHAHEEN (D NH)	CASEY (D PA)
KLOBUCHAR (D MN)	KLOBUCHAR (D MN)
CASEY (D PA)	TESTER (D MT)
FEINSTEIN (D CA)	SHAHEEN (D NH)
KAINE (D VA)	NELSON (D FL)
NELSON (D FL)	CARPER (D DE)
BENNET (D CO)	KAINE (D VA)
CARPER (D DE)	FEINSTEIN (D CA)
TESTER (D MT)	BENNET (D CO)
WARNER (D VA)	KING (Indep ME)
KING (Indep ME)	WARNER (D VA)
MCCASKILL (D MO)	MCCASKILL (D MO)
HEITKAMP (D ND)	HEITKAMP (D ND)
DONNELLY (D IN)	DONNELLY (D IN)
MANCHIN (D WV)	MANCHIN (D WV)
COLLINS (R ME)	COLLINS (R ME)
KIRK (R IL)	KIRK (R IL)
AYOTTE (R NH)	AYOTTE (R NH)
MURKOWSKI (R AK)	GRAHAM (R SC)
GRAHAM (R SC)	MURKOWSKI (R AK)
ALEXANDER (R TN)	PORTMAN (R OH)
PORTMAN (R OH)	ALEXANDER (R TN)
HELLER (R NV)	CAPITO (R WV)
MCCONNELL (R KY)	COCHRAN (R MS)
CAPITO (R WV)	HATCH (R UT)
GARDNER (R CO)	ROUNDS (R SD)
COCHRAN (R MS)	BURR (R NC)
HATCH (R UT)	MCCAIN (R AZ)
CORKER (R TN)	ISAKSON (R GA)
ROUNDS (R SD)	JOHNSON (R WI)
MCCAIN (R AZ)	MCCONNELL (R KY)
BURR (R NC)	TILLIS (R NC)
TILLIS (R NC)	WICKER (R MS)
JOHNSON (R WI)	CORKER (R TN)
HOEVEN (R ND)	HOEVEN (R ND)
ISAKSON (R GA)	CORNYN (R TX)

buds	cmds
CORNYN (R TX)	THUNE (R SD)
THUNE (R SD)	GARDNER (R CO)
GRASSLEY (R IA)	COATS (R IN)
WICKER (R MS)	BLUNT (R MO)
SULLIVAN (R AK)	ROBERTS (R KS)
ENZI (R WY)	CASSIDY (R LA)
COATS (R IN)	GRASSLEY (R IA)
CASSIDY (R LA)	BARASSO (R WY)
ROBERTS (R KS)	ERNST (R IA)
ERNST (R IA)	ENZI (R WY)
BARASSO (R WY)	HELLER (R NV)
FISCHER (R NE)	SULLIVAN (R AK)
BLUNT (R MO)	FISCHER (R NE)
FLAKE (R AZ)	BOOZMAN (R AR)
BOOZMAN (R AR)	FLAKE (R AZ)
INHOFE (R OK)	INHOFE (R OK)
DAINES (R MT)	MORAN (R KS)
MORAN (R KS)	TOOMEY (R PA)
TOOMEY (R PA)	PERDUE (R GA)
PERDUE (R GA)	VITTER (R LA)
SCOTT (R SC)	DAINES (R MT)
LANKFORD (R OK)	LANKFORD (R OK)
COTTON (R AR)	COTTON (R AR)
RUBIO (R FL)	SCOTT (R SC)
CRAPO (R ID)	SESSIONS (R AL)
RISCH (R ID)	RISCH (R ID)
SESSIONS (R AL)	CRAPO (R ID)
VITTER (R LA)	SHELBY (R AL)
SHELBY (R AL)	RUBIO (R FL)
PAUL (R KY)	SASSE (R NE)
CRUZ (R TX)	CRUZ (R TX)
LEE (R UT)	PAUL (R KY)
SASSE (R NE)	LEE (R UT)

```r
sessionInfo()
```

```
## R version 3.4.0 (2017-04-21)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 16.04.1 LTS
##
## Matrix products: default
## BLAS: /usr/lib/libblas/libblas.so.3.6.0
## LAPACK: /usr/lib/lapack/liblapack.so.3.6.0
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
```

```
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] viridis_0.4.0        viridisLite_0.2.0    rstan_2.15.1
##  [4] StanHeaders_2.15.0-1 MCMCglmm_2.24        ape_4.1
##  [7] Matrix_1.2-10        ggplot2_2.2.1        dplyr_0.5.0
## [10] coda_0.19-1          buds_1.0             Rcpp_0.12.11
## [13] knitr_1.15.1
##
## loaded via a namespace (and not attached):
##  [1] highr_0.6        compiler_3.4.0   plyr_1.8.4       tools_3.4.0
##  [5] digest_0.6.12    evaluate_0.10    tibble_1.3.0     gtable_0.2.0
##  [9] nlme_3.1-124     lattice_0.20-33  DBI_0.6-1        yaml_2.1.14
## [13] parallel_3.4.0   gridExtra_2.2.1  stringr_1.2.0    stats4_3.4.0
## [17] rprojroot_1.2    grid_3.4.0       inline_0.3.14    R6_2.2.1
## [21] rmarkdown_1.5    tensorA_0.36     reshape2_1.4.2   corpcor_1.6.9
## [25] magrittr_1.5     backports_1.0.5  scales_0.4.1     codetools_0.2-14
## [29] htmltools_0.3.6  assertthat_0.2.0 princurve_1.1-12 cubature_1.3-8
## [33] colorspace_1.3-2 labeling_0.3     stringi_1.1.5    lazyeval_0.2.0
## [37] munsell_0.4.3
```