Problem Set 1

Applied Stats/Quant Methods 1
Shelly Veal-Upham
25337422

Due: January 28, 2026

Instructions

- Please show your work! You may lose points by simply writing in the answer. If the problem requires you to execute commands in `R`, please include the code you used to get your answers. Please also include the `.R` file that contains your code. If you are not sure if work needs to be shown for a particular problem, please ask.
- Your homework should be submitted electronically on GitHub.
- This problem set is due before 23:59 on Wednesday January 28, 2026. No late assignments will be accepted.

Roll Call Votes in the European Parliament

Data Manipulation

First, you need to download data from the first six elected European Parliaments on each MEP and how they voted in each recorded roll-call vote.

1. Load these datasets into your global environment:
   - `mep_info_26Jul11.xls` (MEP characteristics, EP1–EP5)
   - `rcv_ep1.txt` (EP1 roll-call votes)
2. Briefly describe (2–3 sentences each) the unit of analysis and key variables in each of these two datasets. The EP 1 dataset gives us information on the voting decisions made by each MEP in the first European Parliament (0 = Absent, 1 = Yes, 2 = No, 3 = Abstain, 4 = Present but did not vote, 5 = Absent). It also includes MEP

ID numbers, National Party codes, Member State codes, and EP Group codes. The MEP info dataset gives us some duplicate information as the EP 1 dataset (MEP IDs, National Party, EP Group, and Member State affiliations) along with the Nominate coordinates (-1, 1).

3. The `rcv_ep1` data are in a wide format, with V1, V2, . . . , Vn as separate vote columns.

    - Identify which columns are ID/metadata (*MEPID, MEPNAME, MS, NP, EPG*) and which columns are vote decisions ($V_1 \ldots V_n$). Tidy the voting data such that each row/observation is a single vote for a single MEP.

```r
ep1_rcv <- read_csv('rcv_ep1.txt')
mep_info <- read_csv('mep_info_26Jul11.csv')

ep1_long <- ep1_rcv %>% #converting to long format-- each row is a single vote
  pivot_longer(
    cols = starts_with("V"),
    names_to = "VOTE_ID",
    values_to = "VOTE_CAST"
  )

ep1_rcv <- ep1_rcv %>% select(where(~!all(is.na(.)))) # removing any empty cols
```

    - Create a summary table of counts of decision categories (e.g. Yes/No/Abstain/Present but did not vote/Absent) across all votes.

```r
vote_counts <- table(ep1_long$VOTE_CAST) # compiling all votes
names(vote_counts) <- c("Absent", "Yes", "No", "Abstain",
                        "Present but did not vote",
                        "Non-MEP") #re-labeling vote types
vote_counts <- vote_counts[names(vote_counts) != "Non-MEP"] # removing non-MEPs

vote_counts_df <- as.data.frame(vote_counts)
colnames(vote_counts_df) <- c("Vote_Type", "Count")

xtable(vote_counts_df, caption = "Total Vote Counts for EP 1")
```

	Vote_Type	Count
1	Absent	99753
2	Yes	88185
3	No	75171
4	Abstain	9577
5	Present but did not vote	109224

Table 1: Total Vote Counts for EP 1

4. Construct a new dataset that combines MEP-level information with their vote decisions from EP1 in long format (from part 3). Check for missingness.

```r
mep_info <- mep_info %>% select(where(~!all(is.na(.)))) # removing empty columns
names(mep_info)[1] <- "MEPID" # making MEP id column name identical

merged_df <- merge(mep_info, ep1_rcv, by = "MEPID", all = TRUE)
merged_df <- merged_df %>% mutate(across(c(`NOM-D1`, `NOM-D2`), as.numeric))

w_out_dims <- merged_df[-c(6,7)] #check for missingness (except in na-heavy cols)
rows_w_nas <- w_out_dims[!complete.cases(w_out_dims), ]
nrow(rows_w_nas)

problem_rows <- as.numeric(rownames(rows_w_nas))
print(problem_rows)
```

When we first merge the datasets, we find that the Coordinate columns contain lots of NA values due to that information being missing from the EP 1 dataset. To check for legitimate missingness in the data, I removed those columns and had a look at which rows are missing information elsewhere. I found that rows 45, 444, 493 and 543 had NA's due to information not contained in the EP 1 dataset that weren't in the MEP Info dataset, and row 470 was the sole row with information coming from EP 1 that wasn't in the MEP info dataset. Row 45 refers to Gustavo Selva, whose information I copied from the EP dataset columns to the MEP info dataset columns here:

```r
merged_df[problem_rows, c(8:11)] <- merged_df[problem_rows, c(2:5)]
```

Then, after making a dataset with NA's set to 99, for each of the duplicate columns (MEPID, MS, EPG, and Names) I checked for any other discrepancies before deleting the extra columns.

```r
merged_df_99 <- merged_df[-2]
merged_df_99[is.na(merged_df_99)] <- 99
non_identicals <- merged_df_99[merged_df_99$EPG != merged_df_99$`EP Group`, ]
print(non_identicals) # checking for all the important cols,
#                       none had any rows except Names

# now we are just going to drop ep1 cols, as they're identical
merged_df <- merged_df %>% select(!MEPNAME:EPG)
```

5. Compute, for each EP group in EP1:

- The mean rate of Yes votes (Yes over Yes+No+Abstain) across all roll calls.
- The mean abstention rate.
- The mean vote preferences along the two contested dimensions (NOM-D1 and NOM-D2).

	0	1	2	3	4	5	YESPROP	ABSPROP
C	9547	14421	17707	2612	11531	0	0.42	0.08
E	22917	25909	23914	1093	28218	16673	0.51	0.02
G	5624	3436	2806	468	7328	21094	0.51	0.07
L	11046	6129	5679	797	10903	7088	0.49	0.06
M	14739	6726	4990	1019	14005	5479	0.53	0.08
N	2976	1997	1247	193	3857	11880	0.58	0.06
R	3946	926	562	537	3779	1768	0.46	0.27
S	28955	28641	18266	2858	29603	38753	0.58	0.06

```r
df_long <- merged_df %>% #converting to long format-- each row is a
    single vote
  pivot_longer(
    cols = starts_with("V"),
    names_to = "VOTE_ID",
    values_to = "VOTE_CAST"
  )

vote_sums <- as.data.frame.matrix(table(df_long$`EP Group`, df_long$
    VOTE_CAST))

vote_sums$YESPROP <- vote_sums$`1`/(vote_sums$`1` +
                                     vote_sums$`2` +
                                     vote_sums$`3`)

vote_sums$ABSPROP <- vote_sums$`3`/(vote_sums$`1` +
                                     vote_sums$`2` +
                                     vote_sums$`3`)

tapply(df_long$VOTE_CAST, list(df_long$`EP Group`,
                               df_long$`NOM-D1`,
                               df_long$`NOM-D2`), mean)

xtable(vote_sums)

sel <- !is.na(df_long$VOTE_CAST) &
  df_long$VOTE_CAST >= 1 &
  df_long$VOTE_CAST <= 3

d1_averages <- tapply(
  df_long$VOTE_CAST[sel],
  list(
    df_long$`EP Group`[sel],
    df_long$`NOM-D1`[sel]
  ),
  mean
)

d2_averages <- tapply(
```

```
  df_long$VOTE_CAST[sel],
  list(
    df_long$`EP Group`[sel],
    df_long$`NOM-D2`[sel]
  ),
  mean
)
```

Data Visualization

1. Plot the distribution of the first NOMINATE dimension by EP group, and explain any trends you see.

Figure 1: Nominate Dimension 1.

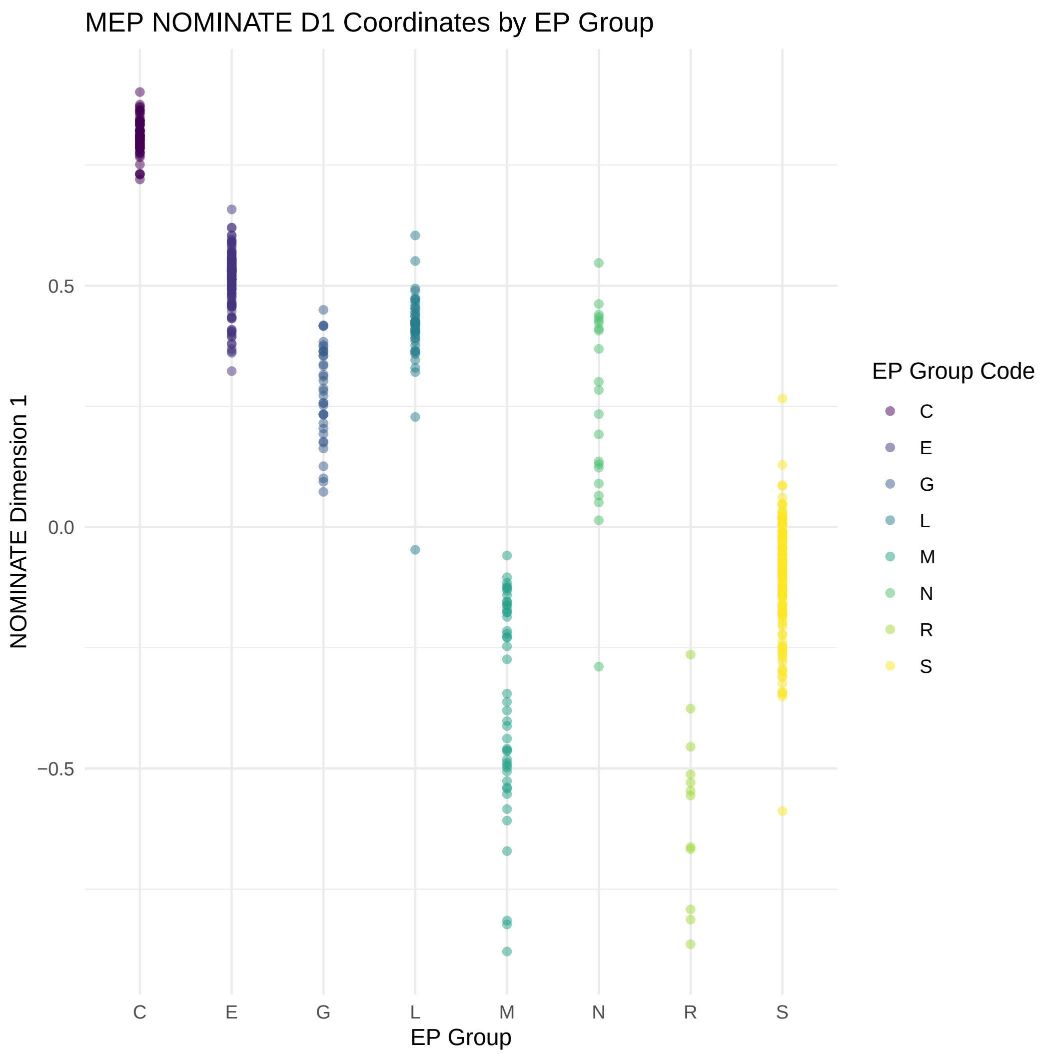
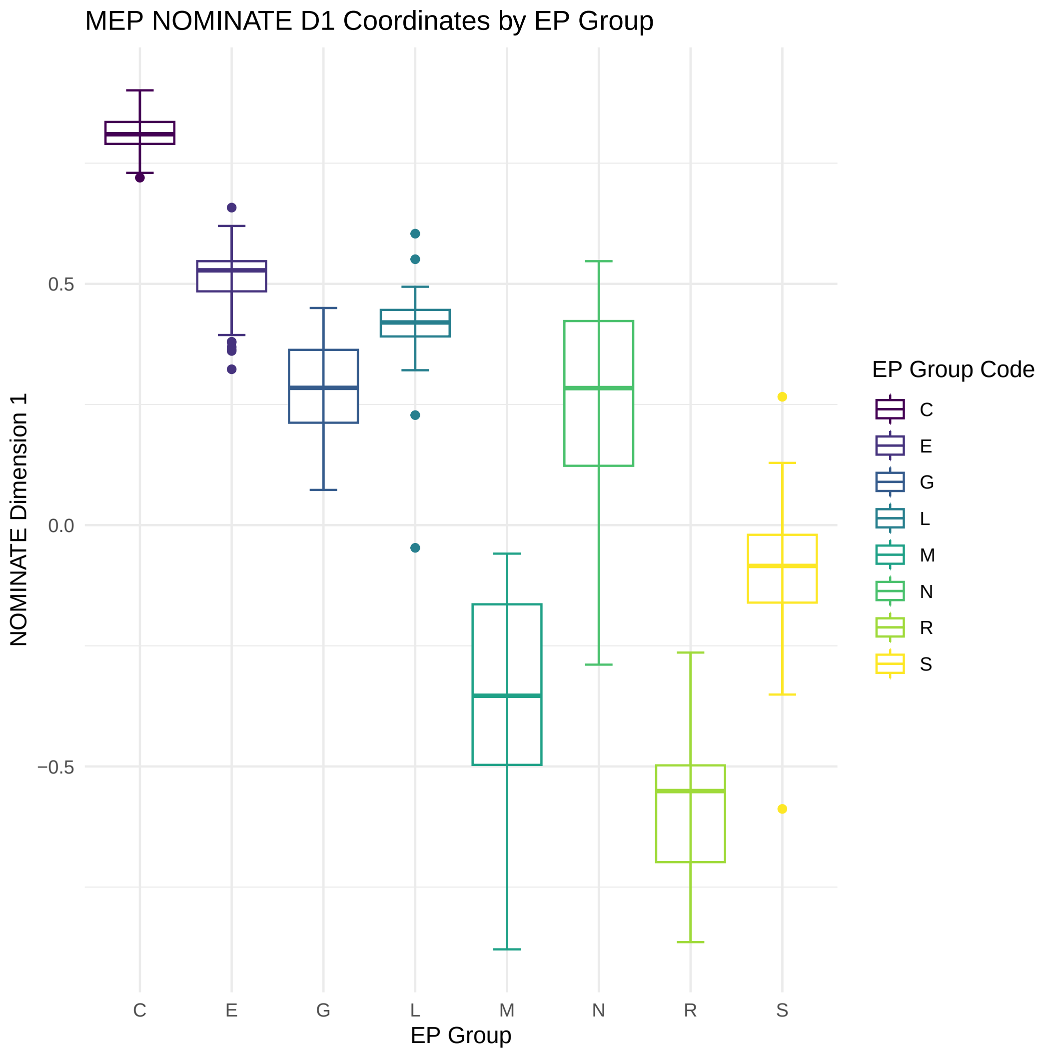


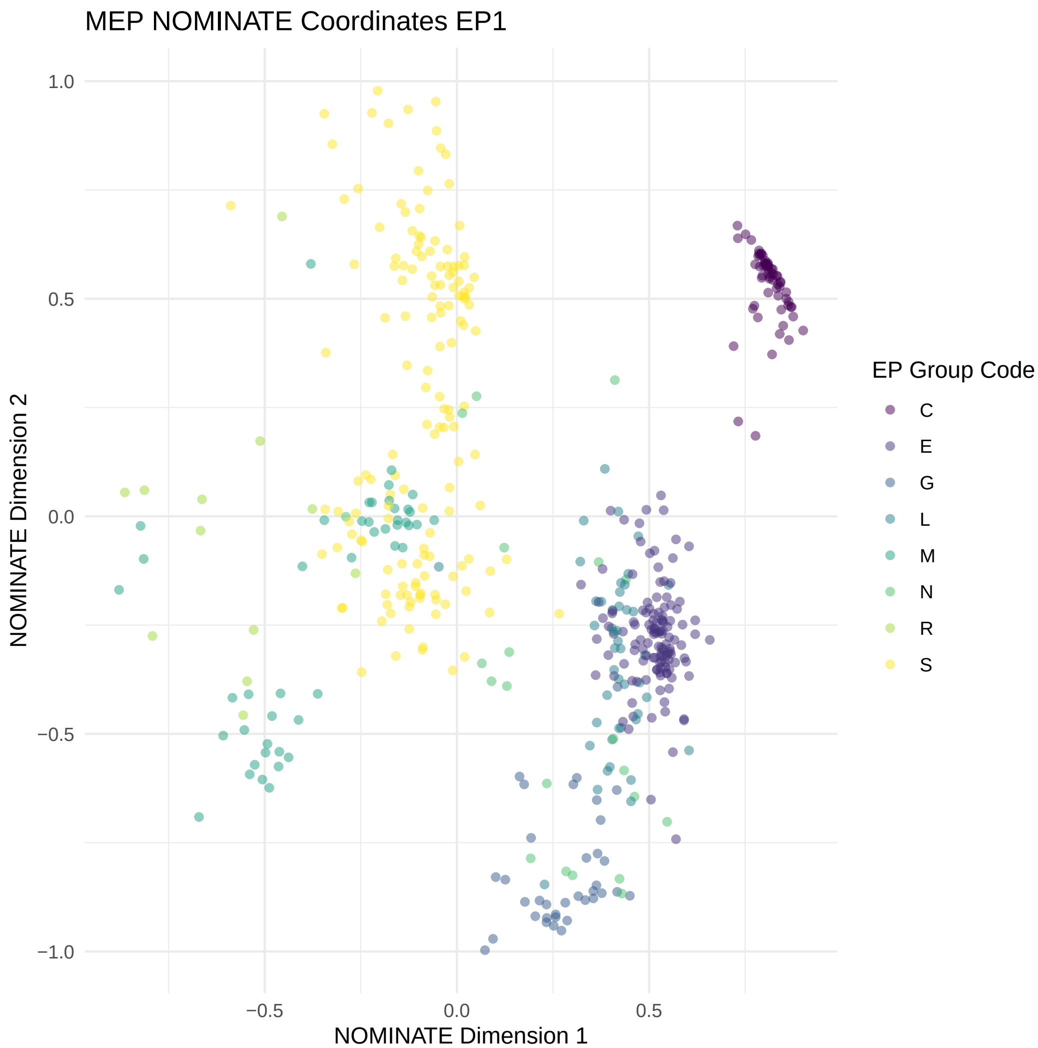
Figure 2: Nominate Dimension 1 (2).



We can see that EP groups are generally in the same areas as one another with regard to dimension 1, which makes sense. Groups M, N, and R seem to have the most spread.

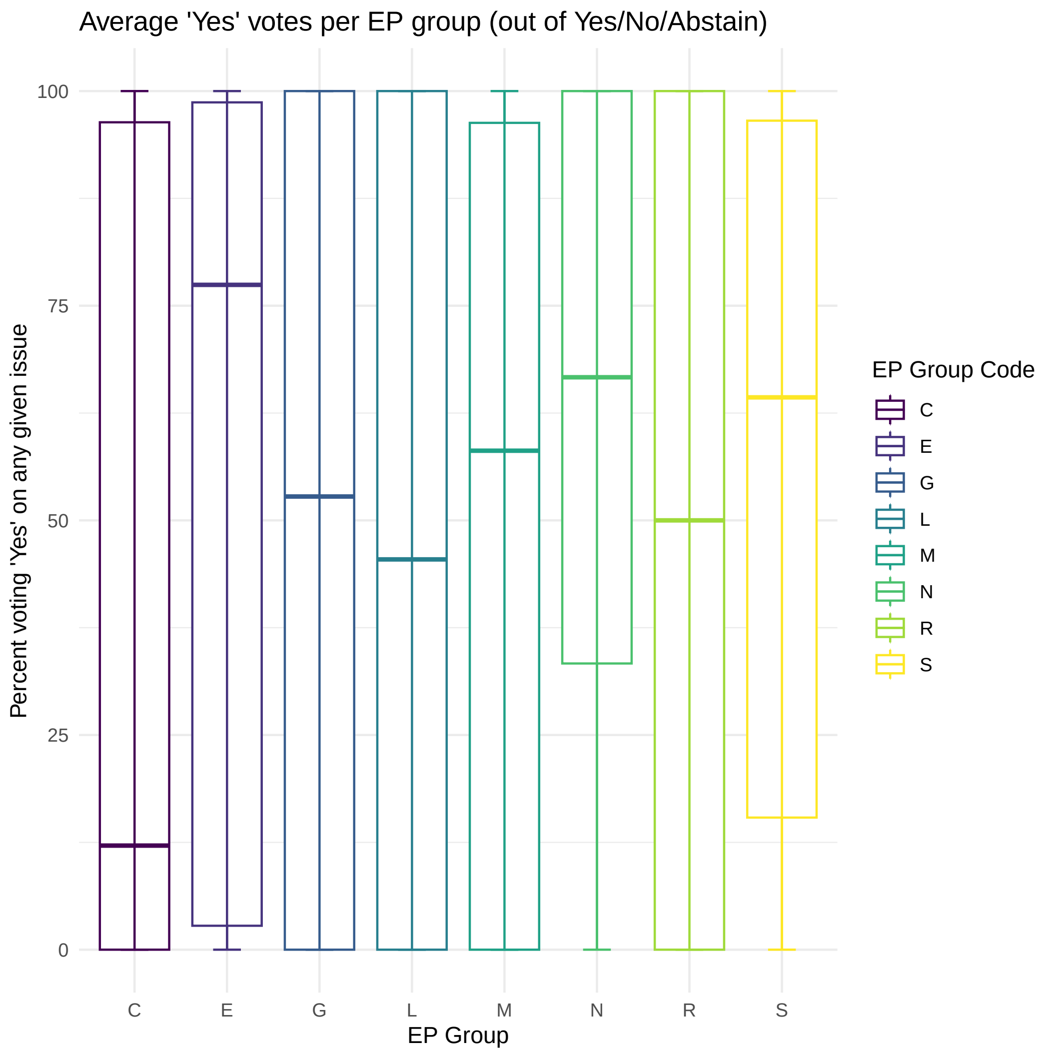
2. Make a scatterplot of *nomdim1* (x-axis) and *nomdim2* (y-axis), with one point per MEP and color by EP group.

Figure 3: Nominate Dimensions 1 and 2.



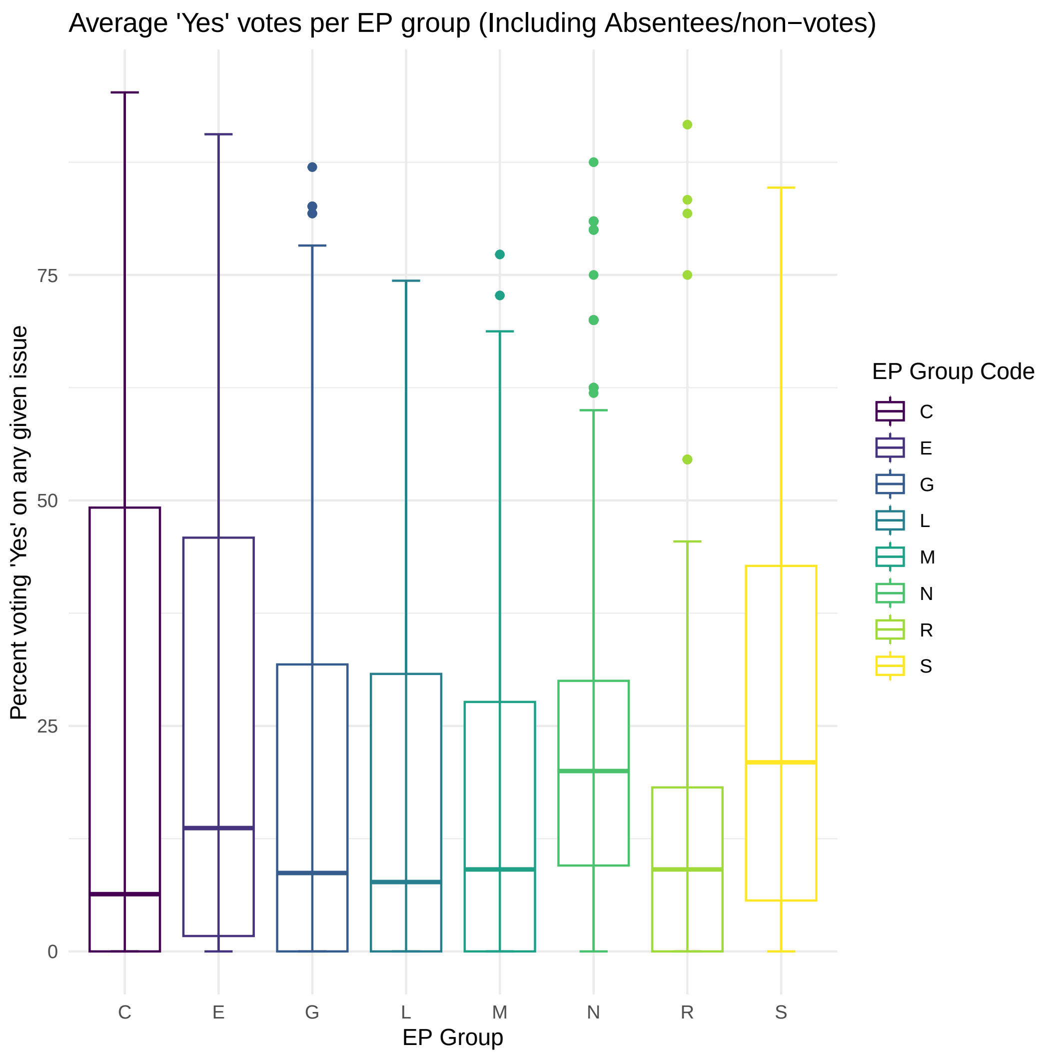
3. Produce a boxplot of the proportion voting *Yes* by EP group to visualize cohesion.

Figure 4: Proportion of Yeses (Only out of Yes/No/Abstains).



This plot seemed a little odd / underinformative to me, so I included all vote types in the following plot as well:

Figure 5: Proportion of Yeses (including Absent, Didn't vote).



4. Display the proportion voting *Yes* by national party using a bar plot.

Figure 6: Proportion of Yeses by National Party.