Problem Set 1

Data Visualisation for Social Scientists

Due: January 28, 2026

Roll Call Votes in the European Parliament

Data Manipulation

1. Load these datasets into your global environment:

   - `mep_info_26Jul11.xls` (MEP characteristics, EP1–EP5)
   - `rcv_ep1.txt` (EP1 roll-call votes)

   I load the EP1 roll-call vote data and the EP1 MEP information at the start of the workflow so that every step is reproducible from the original source files.

```r
# Load EP1 roll-call votes (wide format)
rcv_ep1 <- read_csv("rcv_ep1.txt", show_col_types = FALSE)
```

```r
# Load MEP-level information (EP1 sheet is sheet = 2)
mep_info <- read_excel("mep_info_26Jul11.xls", sheet = 2) %>%
  rename(MEPID = `MEP id`)
```

2. Briefly describe (2–3 sentences each) the unit of analysis and key variables in each of these two datasets.

   MEP information dataset (`mep_info_26Jul11.xls`). The unit of analysis is the individual Member of the European Parliament (MEP). Each row contains one MEP's identifying information, party/group affiliation, and ideological coordinates. Key variables include `MEPID`, `MS`, `NP`, `EP Group`, and the NOMINATE dimensions `NOM-D1` and `NOM-D2`.

   Roll-call vote dataset (`rcv_ep1.txt`). In the original wide format, each row corresponds to one MEP and each roll-call vote is stored in separate columns (`V1`–`Vn`). After reshaping into long format, the unit of analysis becomes a single MEP's vote on a single roll-call vote. Key variables include `MEPID`, `MEPNAME`, `MS`, `NP`, `EPG`, and the vote columns `V1`–`Vn`.

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

   - Identify which columns are ID/metadata and which columns are vote decisions. Tidy the data so that each row is a single vote by a single MEP.
   - Create a summary table of counts of decision categories across all votes.

   I first separate ID/metadata variables (`MEPID, MEPNAME, MS, NP, EPG`) from the roll-call vote variables (`V1`–`Vn`). I then reshape the dataset from wide to long format using `pivot_longer()` so that each row corresponds to one MEP–rollcall vote. Finally, I recode the numeric vote codes into substantive categories and compute category counts.

```r
# Identify ID/metadata columns and vote decision columns
id_cols <- c("MEPID", "MEPNAME", "MS", "NP", "EPG")
vote_cols <- names(rcv_ep1) %>% str_subset("^V\\d+$")  # V1, V2, ..., Vn

# Check
print(setdiff(id_cols, names(rcv_ep1)))
print(length(vote_cols))

# Tidy voting data: wide -> long (one row = one MEP x one roll-call)
rcv_ep1_long <- rcv_ep1 %>%
  pivot_longer(
    cols = all_of(vote_cols),
    names_to = "rollcall_id",
    values_to = "vote_code"
  ) %>%
  mutate(
    decision = case_when(
      vote_code == 1 ~ "Yes",
      vote_code == 2 ~ "No",
      vote_code == 3 ~ "Abstain",
      vote_code == 4 ~ "Present but did not vote",
      vote_code == 0 ~ "Absent",
      vote_code == 5 ~ "Not an MEP",
      TRUE ~ "Other/Unknown"
    )
  )

# Summary table of decision categories across all votes
decision_counts <- rcv_ep1_long %>%
  count(decision, sort = TRUE)
print(decision_counts)
```

Output (decision category counts).

```
decision                       n
<chr>                      <int>
1 Present but did not vote 109224
2 Not an MEP               103618
```

```
3 Absent                    99753
4 Yes                       88185
5 No                        75171
6 Abstain                    9577
```

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.

   I merge the long-format roll-call votes with the MEP-level information by `MEPID`. After merging, I summarize missingness in key variables to verify data quality (especially the NOMINATE dimensions and EP group).

```r
# Merge MEP information with voting data (by MEPID)
rcv_ep1_merged <- rcv_ep1_long %>%
  left_join(mep_info, by = "MEPID")

# Check missingness after merge (key variables)
missing_summary <- rcv_ep1_merged %>%
  summarise(
    missing_nomd1 = sum(is.na(`NOM-D1`)),
    missing_nomd2 = sum(is.na(`NOM-D2`)),
    missing_epg   = sum(is.na(`EP Group`))
  )
print(missing_summary)
```

   Output (missingness summary).

```
missing_nomd1 missing_nomd2 missing_epg
<int>         <int>       <int>
1            886           886         886
```

5. Compute, for each EP group in EP1:

6. 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 NOM-D1 and NOM-D2.

   I restrict the data to valid voting decisions (Yes/No/Abstain; vote codes 1–3). For each EP group, I compute (i) the Yes rate as the mean of the indicator $\mathbb{1}(\text{vote} = Yes)$, (ii) the abstention rate as the mean of $\mathbb{1}(\text{vote} = Abstain)$, and (iii) the mean values of `NOM-D1` and `NOM-D2`. Because the NOMINATE variables contain "." entries, I first recode them as missing and then convert them to numeric.

```r
# Convert "." to NA and cast NOMINATE dimensions to numeric
rcv_ep1_merged <- rcv_ep1_merged %>%
  mutate(
    `NOM-D1` = na_if(as.character(`NOM-D1`), "."),
```

```r
    `NOM-D2` = na_if(as.character(`NOM-D2`), "."),
    `NOM-D1` = as.numeric(`NOM-D1`),
    `NOM-D2` = as.numeric(`NOM-D2`)
  )

ep_group_summary <- rcv_ep1_merged %>%
  filter(vote_code %in% c(1, 2, 3)) %>%   # valid votes: Yes/No/Abstain
  group_by(`EP Group`) %>%
  summarise(
    yes_rate = mean(vote_code == 1, na.rm = TRUE),
    abstention_rate = mean(vote_code == 3, na.rm = TRUE),
    mean_nomdim1 = mean(`NOM-D1`, na.rm = TRUE),
    mean_nomdim2 = mean(`NOM-D2`, na.rm = TRUE),
    .groups = "drop"
  )
print(ep_group_summary)
```

Results (from the EP-group summary table).

(a) **Mean Yes rate.** The highest Yes rate is for EP Group N (0.581), followed by S (0.576) and M (0.528). The lowest Yes rate is for EP Group C (0.415).

(b) **Mean abstention rate.** EP Group R has the highest abstention rate (0.265). All other groups have abstention rates below 0.10, with the lowest for EP Group E (0.0215).

(c) **Mean NOMINATE preferences (NOM-D1 and NOM-D2).** EP Group C has the highest mean NOM-D1 (0.811), whereas EP Group R has the lowest mean NOM-D1 (-0.586). For NOM-D2, EP Group C has the highest mean (0.531) and EP Group G has the lowest mean (-0.817).

Full output (EP-group summary table).

```
  `EP Group` yes_rate abstention_rate mean_nomdim1 mean_nomdim2
  <chr>         <dbl>           <dbl>        <dbl>        <dbl>
1 C             0.415          0.0752       0.811        0.531
2 E             0.509          0.0215       0.513       -0.268
3 G             0.512          0.0697       0.289       -0.817
4 L             0.486          0.0632       0.420       -0.301
5 M             0.528          0.0800      -0.299       -0.149
6 N             0.581          0.0562       0.202       -0.195
7 R             0.457          0.265       -0.586       -0.0869
8 S             0.576          0.0574      -0.0907       0.390
```

Data Visualization

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

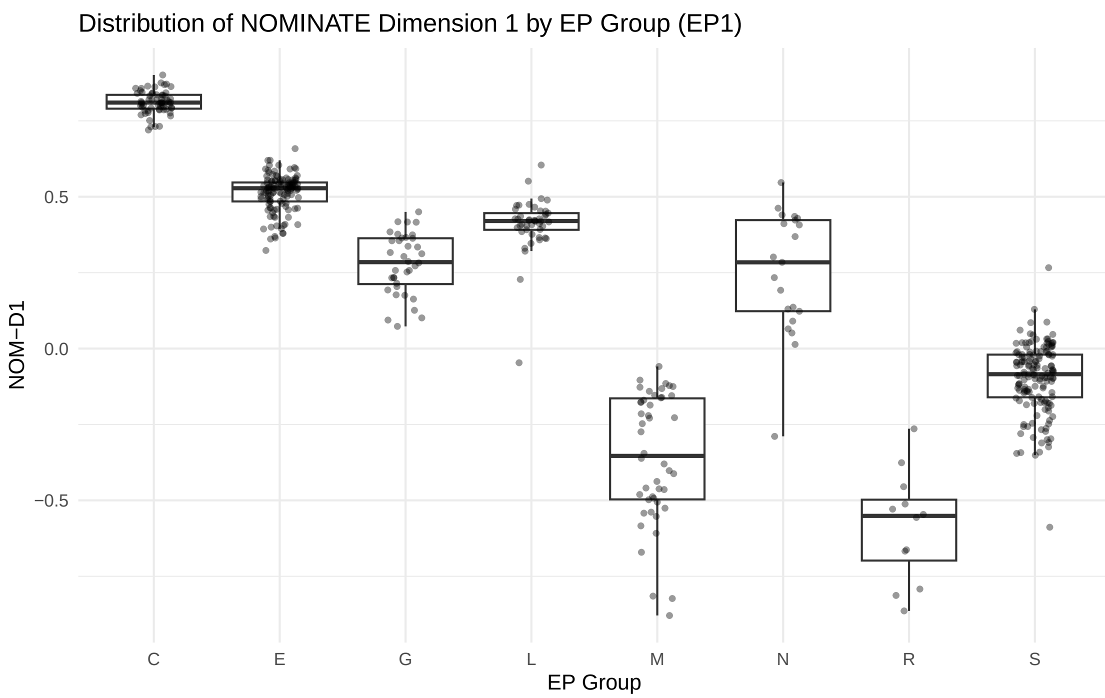

Figure 1: Distribution of NOMINATE Dimension 1 by EP Group (EP1).

The boxplots show clear differences in the median and spread of NOMINATE Dimension 1 across EP groups. For example, EP Group C is concentrated at higher NOM-D1 values, while EP Group R is centered on lower values, indicating ideological separation.

```
mep_ep1 <- rcv_ep1_merged %>%
  distinct(MEPID, `EP Group`, `NOM-D1`, `NOM-D2`) %>%
  filter(!is.na(`NOM-D1`))

pdf("viz1_SK.pdf", width = 8, height = 5)
ggplot(mep_ep1, aes(x = `EP Group`, y = `NOM-D1`)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(width = 0.15, alpha = 0.4, size = 1) +
  labs(
    title = "Distribution of NOMINATE Dimension 1 by EP Group (EP1)",
    x = "EP Group",
    y = "NOM-D1"
  ) +
  theme_minimal()
dev.off()
```

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

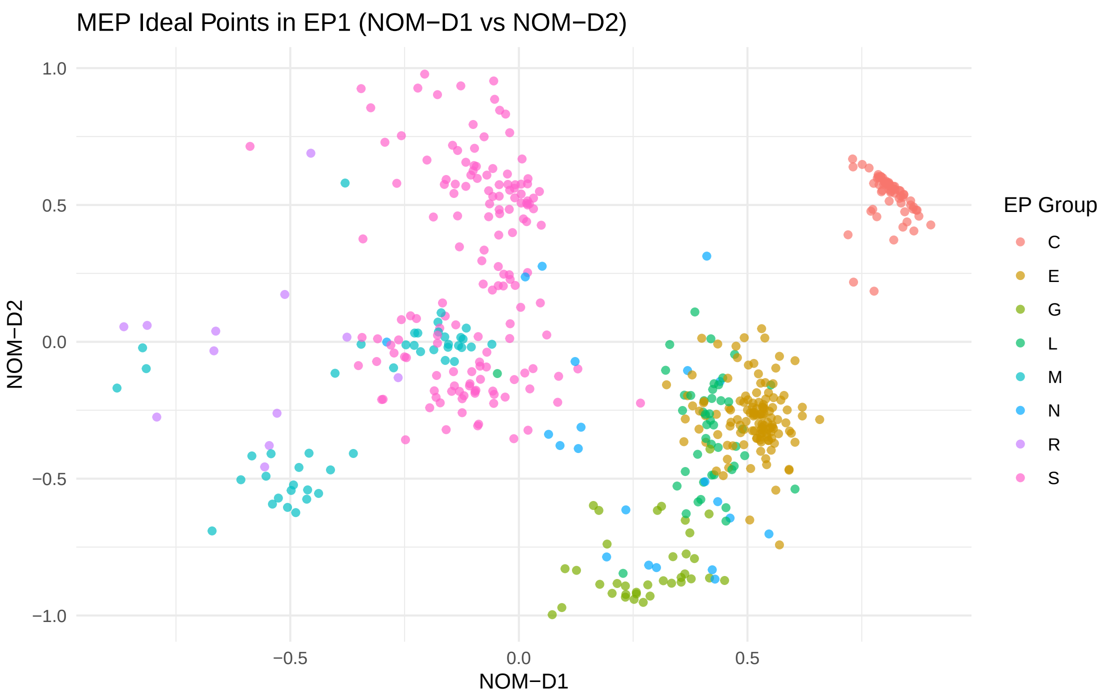

Figure 2: MEP ideal points on NOMINATE Dimensions 1 and 2, colored by EP Group.

MEPs cluster by EP group in the two-dimensional ideological space. While many groups occupy distinct regions, partial overlap remains, which may reflect ideological proximity between groups or within-group heterogeneity.

```r
mep_ep1_scatter <- rcv_ep1_merged %>%
  distinct(MEPID, `EP Group`, `NOM-D1`, `NOM-D2`) %>%
  filter(!is.na(`NOM-D1`), !is.na(`NOM-D2`))

pdf("viz2_SK.pdf", width = 8, height = 5)
ggplot(mep_ep1_scatter, aes(x = `NOM-D1`, y = `NOM-D2`, color = `EP Group
   `)) +
  geom_point(alpha = 0.7, size = 1.5) +
  labs(
    title = "MEP Ideal Points in EP1 (NOM-D1 vs NOM-D2)",
    x = "NOM-D1",
    y = "NOM-D2",
    color = "EP Group"
  ) +
  theme_minimal()
dev.off()
```

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

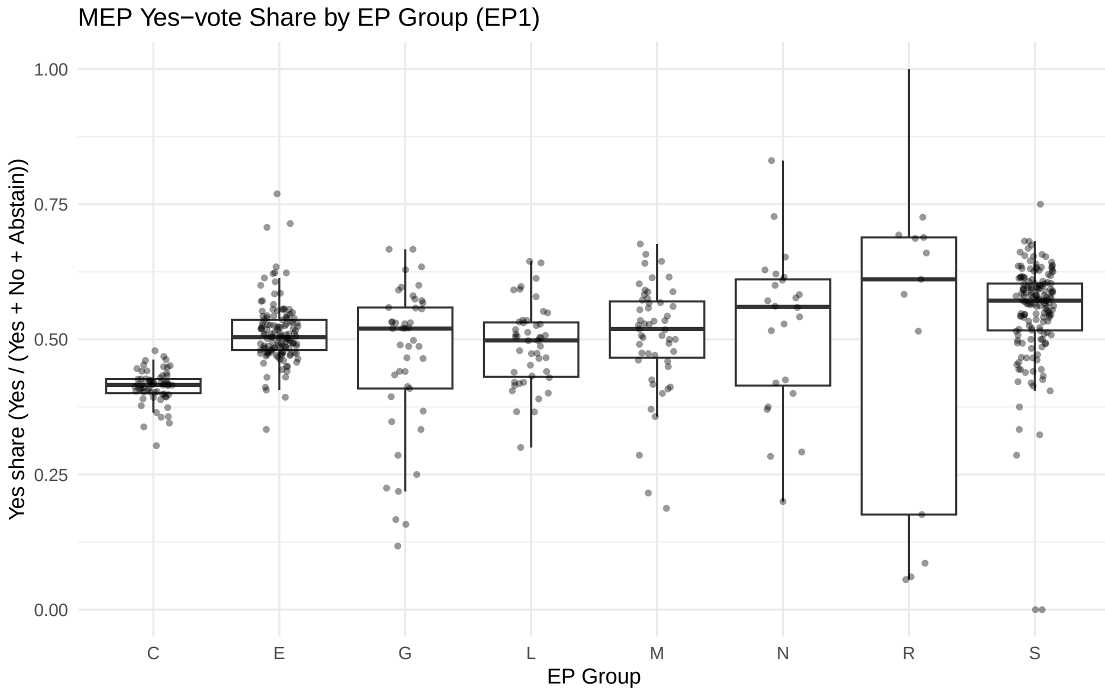

Figure 3: Distribution of MEP-level Yes-vote shares by EP Group (EP1).

Groups with narrower distributions of individual-level Yes shares exhibit higher internal cohesion, whereas wider distributions indicate greater variation in voting behavior within the group.

```
mep_yes_share <- rcv_ep1_merged %>%
  filter(vote_code %in% c(1, 2, 3)) %>%
  group_by(MEPID, `EP Group`) %>%
  summarise(
    yes_share = mean(vote_code == 1, na.rm = TRUE),
    n_votes = n(),
    .groups = "drop"
  )

pdf("viz3_SK.pdf", width = 8, height = 5)
ggplot(mep_yes_share, aes(x = `EP Group`, y = yes_share)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(width = 0.15, alpha = 0.4, size = 1) +
  scale_y_continuous(limits = c(0, 1)) +
  labs(
    title = "MEP Yes-vote Share by EP Group (EP1)",
    x = "EP Group",
    y = "Yes share (Yes / (Yes + No + Abstain))"
  ) +
```

```
  theme_minimal()
dev.off()
```

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

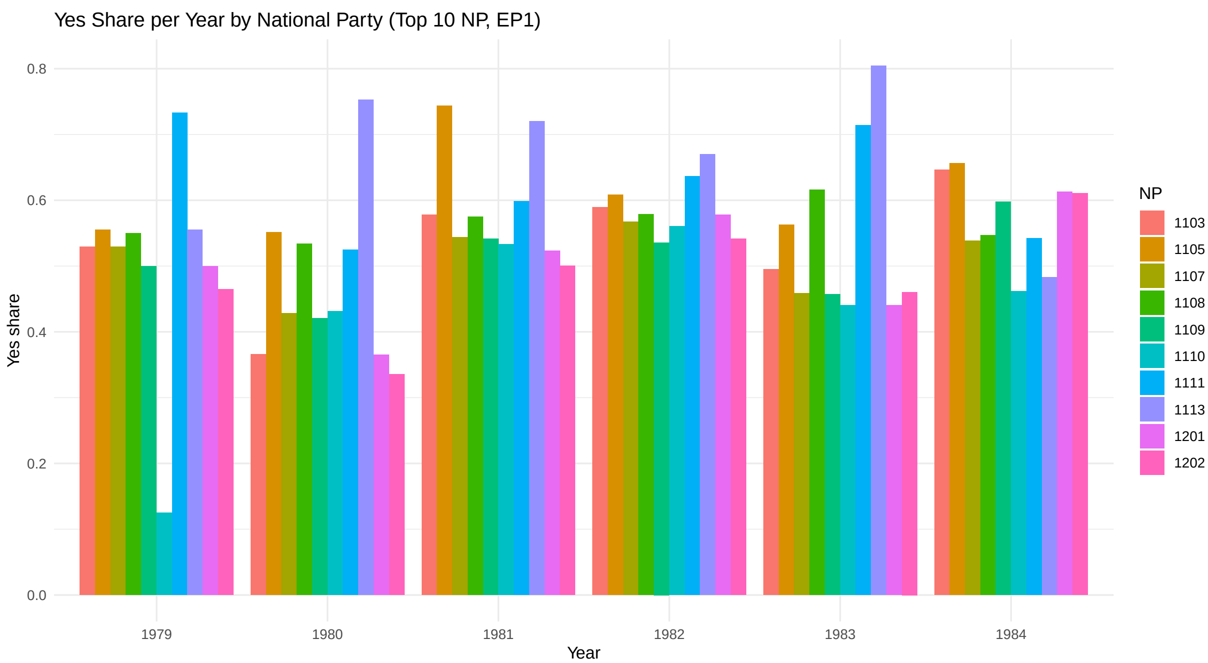

Figure 4: Yes-vote share per year by national party (Top 10 NP, EP1).

I use the roll-call metadata file to extract the year of each roll-call vote and merge it onto the long-format vote data. The date variable is stored as an Excel serial number, so I convert it to a calendar date using `as.Date(., origin = "1899-12-30")` before extracting the year. I then compute average Yes shares by year and national party and display them in a bar plot (restricted to the top 10 national parties for readability). Some roll-calls have missing date information in the metadata; those are excluded from the year-based plots.

```
vote_info <- read_excel("vote_info_Jun2010.xls", sheet = "EP1")

vote_info_ep1 <- vote_info %>%
  filter(`EP No.` == 1) %>%
  mutate(
    vote_no = as.integer(`Vote No. in RCV_EP1 file`),
    # Date is stored as Excel serial number (character)
    date = as.Date(as.numeric(Date), origin = "1899-12-30"),
    year = format(date, "%Y")
  ) %>%
  select(vote_no, year) %>%
  filter(!is.na(year))   # drop roll-calls with missing dates
```

```r

# Merge year onto rcv long data
rcv_ep1_long_year <- rcv_ep1_long %>%
  mutate(vote_no = as.integer(sub("^V", "", rollcall_id))) %>%
  left_join(vote_info_ep1, by = "vote_no")

# Check remaining missing years
print(sum(is.na(rcv_ep1_long_year$year)))

# Year x NP yes share
np_year_yes <- rcv_ep1_long_year %>%
  filter(vote_code %in% c(1, 2, 3)) %>%
  filter(!is.na(year)) %>%
  group_by(year, NP) %>%
  summarise(yes_share = mean(vote_code == 1), .groups = "drop")

# Keep top 10 national parties for readability
top_np <- np_year_yes %>%
  count(NP, sort = TRUE) %>%
  slice_head(n = 10) %>%
  pull(NP)

np_year_yes_top <- np_year_yes %>%
  filter(NP %in% top_np)

pdf("viz4_SK.pdf", width = 11, height = 6)
ggplot(np_year_yes_top, aes(x = year, y = yes_share, fill = factor(NP)))
    +
  geom_col(position = "dodge") +
  labs(
    title = "Yes Share per Year by National Party (Top 10 NP, EP1)",
    x = "Year",
    y = "Yes share",
    fill = "NP"
  ) +
  theme_minimal()
dev.off()
```

Output (remaining missing years after merge).

```
> print(sum(is.na(rcv_ep1_long_year$year)))
[1] 9864
```

5. For each EP group, calculate the average *Yes* share per year and plot a line graph.

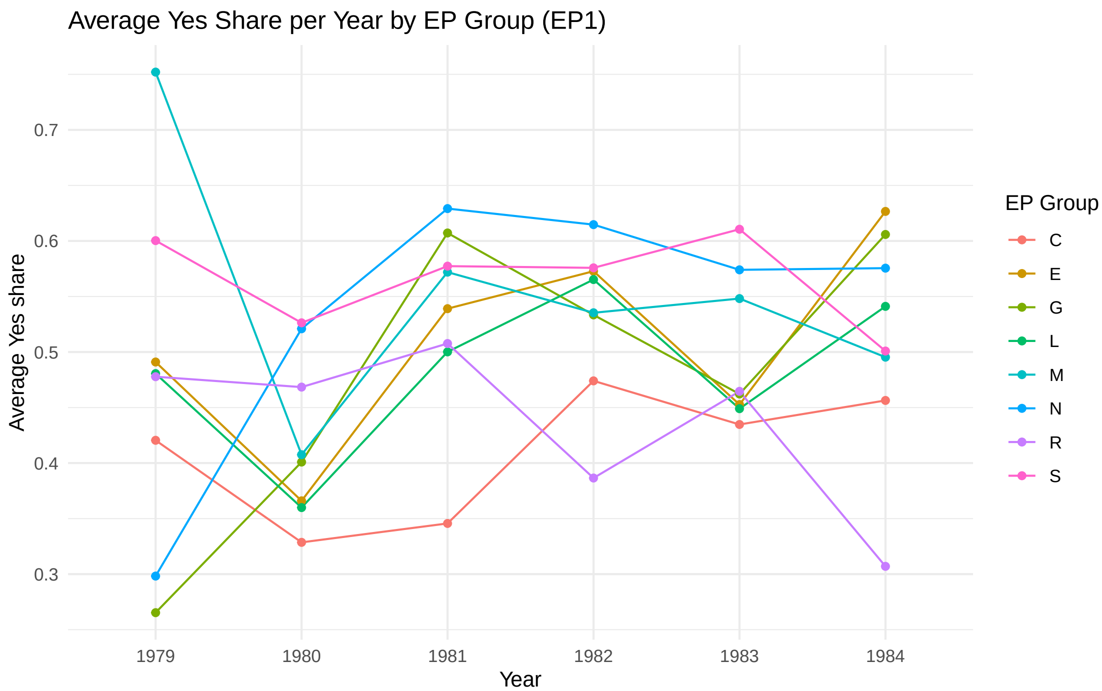

Figure 5: Average Yes-vote share per year by EP Group (EP1).

The line plot shows how average Yes-vote rates change over time for each EP group. While some groups display relatively stable voting patterns across years, others exhibit noticeable fluctuations, suggesting changes in voting behavior across roll calls.

```
rcv_ep1_merged_year <- rcv_ep1_merged %>%
  mutate(vote_no = as.integer(sub("^V", "", rollcall_id))) %>%
  left_join(vote_info_ep1, by = "vote_no")

epg_year_yes <- rcv_ep1_merged_year %>%
  filter(vote_code %in% c(1, 2, 3)) %>%
  filter(!is.na(year)) %>%
  group_by(year, `EP Group`) %>%
  summarise(avg_yes = mean(vote_code == 1), .groups = "drop")

pdf("viz5_SK.pdf", width = 8, height = 5)
ggplot(epg_year_yes, aes(x = year, y = avg_yes,
                         group = `EP Group`, color = `EP Group`)) +
  geom_line() +
  geom_point() +
  labs(
    title = "Average Yes Share per Year by EP Group (EP1)",
    x = "Year",
    y = "Average Yes share",
```

```r
    color = "EP Group"
  ) +
  theme_minimal()
dev.off()

print(head(epg_year_yes, 20))
```

Output (first rows of EP-group-by-year averages).

```
   year  `EP Group` avg_yes
<chr> <chr>        <dbl>
1 1979  C            0.421
2 1979  E            0.491
3 1979  G            0.265
4 1979  L            0.481
5 1979  M            0.752
6 1979  N            0.298
7 1979  R            0.478
8 1979  S            0.600
9 1980  C            0.329
10 1980  E            0.366
11 1980  G            0.401
12 1980  L            0.360
13 1980  M            0.408
14 1980  N            0.521
15 1980  R            0.468
16 1980  S            0.526
17 1981  C            0.346
18 1981  E            0.539
19 1981  G            0.607
20 1981  L            0.500
```