Weeks 5–6: R Fundamentals, Descriptive Statistics & Frequency Distributions

Instructor: Nazik Dinçtopal Deniz | TA: Onur Keleş

Reading. Keith Johnson (2008), *Quantitative Methods in Linguistics* (Blackwell), Chapter 1, especially Sections 1.3–1.7.

Datasets. Put these two files in your working directory:

- `Grammaticality_Ratings.csv`: Johnson's Table 1.2, 36 hypothetical ratings
- `cherokeeVOT.txt`: Voice Onset Time (ms) for Cherokee /t/ and /k/, recorded in 1971 and 2001

1. Working directory and file setup

Check which folder R is looking at. To change it, remove the `#` and replace the path with your own folder.

```
getwd()
# setwd("~/Documents/LING411/Week5_6")
```

2. Grammaticality ratings: import and inspect (discrete / ordinal data)

```
df_grammaticality <- read.csv("Grammaticality_Ratings.csv")

str(df_grammaticality)      # structure: rows, columns, and data types
head(df_grammaticality)     # first 6 rows
summary(df_grammaticality)  # minimum, quartiles, median, mean, maximum
nrow(df_grammaticality)     # number of rows (observations)
names(df_grammaticality)    # column names
```

Use `$` to look inside one column (`dataframe$column`):

```
range(df_grammaticality$rating)   # lowest and highest rating (1 and 9)
length(df_grammaticality$rating)  # number of ratings (36)
anyNA(df_grammaticality$rating)   # any missing values? (FALSE = none)
```

3. Frequency distributions for grammaticality ratings

Johnson's scale runs from 1 (sounds quite ungrammatical) to 10 (sounds perfectly OK), and in his Table 1.3 he tabulates ratings from 1 to 9. Notice that **nobody** gave a rating of 8. If we just run `table()` on the ratings, R skips 8 completely. If we tell R that the possible levels are `1:9` with `factor()`, R keeps 8 in the table with a count of 0.

```
rating_levels <- 1:9

# 3a. Absolute frequencies (raw counts of each rating)
rating_frequency <- table(factor(df_grammaticality$rating, levels = rating_levels))
rating_frequency
sum(rating_frequency)       # check: the total is 36

# 3b. Relative frequencies (proportions from 0 to 1: count / 36)
rating_relative <- prop.table(rating_frequency)
round(rating_relative, 3)   # round to 3 decimal places

# 3c. Percentages (0% to 100%)
rating_percent <- 100 * rating_relative
round(rating_percent, 1)    # round to 1 decimal place

# 3d. Cumulative frequencies and cumulative percentages
#     cumsum() adds up the numbers from rating 1 to rating 9
rating_cumulative         <- cumsum(rating_frequency)
rating_cumulative_percent <- cumsum(rating_percent)
```

Put all the results together in one table (this matches Johnson's Table 1.3):

```
rating_distribution <- data.frame(
  rating                = rating_levels,
  frequency             = as.numeric(rating_frequency),
  relative_freq         = round(as.numeric(rating_relative), 3),
  percentage            = round(as.numeric(rating_percent), 1),
  cumulative_freq       = rating_cumulative,
  cumulative_percentage = round(rating_cumulative_percent, 1)
)
print(rating_distribution, row.names = FALSE)
```

4. Plotting the grammaticality rating distribution

4a. Bar plot. Ratings are separate whole numbers (discrete categories), so a bar plot with spaces between the bars is a good choice.

```
barplot(
  rating_frequency,
  main   = "Frequency Distribution of Grammaticality Ratings",
  xlab   = "Grammaticality Rating (1 = Ungrammatical, 9 = Perfectly OK)",
  ylab   = "Frequency (Count)",
  ylim   = c(0, 14),
  col    = "lightblue",
  border = "navy"
)
```

4b. Connected line plot (in the style of Johnson's Figure 1.2):

```
plot(
  x    = rating_levels,
  y    = as.numeric(rating_frequency),
  type = "b",        # "b" = BOTH points and connecting lines
  pch  = 19,         # 19 = solid circles
  col  = "navy",
  ylim = c(0, 14),
  main = "Grammaticality Rating Frequency Distribution",
  xlab = "Rating",
  ylab = "Frequency"
)
axis(1, at = rating_levels)  # a tick mark for every integer from 1 to 9
```

5. Descriptive statistics for grammaticality ratings

Ratings are ordered categories (ordinal data). Always check the median, mode and IQR alongside the mean and standard deviation.

```
# Central tendency (where is the center?)
mean(df_grammaticality$rating)     # average rating
median(df_grammaticality$rating)   # middle rating when sorted

# Mode (the most frequent rating): which rating has the highest count?
rating_frequency
rating_frequency[rating_frequency == max(rating_frequency)]

# Dispersion (how spread out are the ratings?)
sd(df_grammaticality$rating)       # standard deviation
range(df_grammaticality$rating)    # lowest and highest scores
IQR(df_grammaticality$rating)      # spread of the middle 50%
```

Draw vertical lines for the mean (red) and the median (dark green) on top of the current plot with `abline(v = ...)`:

```
abline(v = mean(df_grammaticality$rating),   col = "red",       lwd = 2, lty = 2)
abline(v = median(df_grammaticality$rating), col = "darkgreen", lwd = 2, lty = 3)
```

6. Cherokee VOT data: import and inspect (continuous acoustic data)

Voice Onset Time (VOT) is a continuous measurement in milliseconds. `cherokeeVOT.txt` is a text file separated by spaces or tabs, so we use `read.table()` instead of `read.csv()`.

```
df_vot <- read.table("cherokeeVOT.txt", header = TRUE)

str(df_vot)
head(df_vot)
summary(df_vot)
names(df_vot)

# How many tokens per year (1971 vs. 2001) and consonant (/k/ vs. /t/)?
table(df_vot$year)
table(df_vot$Consonant)
table(df_vot$year, df_vot$Consonant)
```

7. Cherokee VOT: distributions, subgroups, and normality

7a. Overall descriptive statistics (all data pooled). `na.rm = TRUE` tells R to ignore missing values.

```
mean(df_vot$VOT, na.rm = TRUE)
median(df_vot$VOT, na.rm = TRUE)
sd(df_vot$VOT, na.rm = TRUE)
range(df_vot$VOT, na.rm = TRUE)
IQR(df_vot$VOT, na.rm = TRUE)
```

7b. Overall histogram. For continuous numbers, use `hist()` instead of `barplot()`.

```
hist(
  df_vot$VOT,
  main   = "Overall Distribution of Cherokee VOT (All Data Pooled)",
  xlab   = "Voice Onset Time (ms)",
  ylab   = "Frequency",
  col    = "lightblue",
  border = "navy"
)
```

7c. Overall normal Q-Q plot. If the data follow a normal (bell-shaped) distribution, the dots fall along the straight diagonal line. Dots that curve away from the line show skewness.

```
qqnorm(df_vot$VOT, main = "Normal Q-Q Plot: Pooled Cherokee VOT")
qqline(df_vot$VOT, col = "red", lwd = 2)
```

7d. Breaking the numbers down by group with `aggregate()`. Pooling 1971 and 2001 (or /k/ and /t/) hides important phonetic differences. In R, the tilde ~ means "broken down by".

```
aggregate(VOT ~ year, data = df_vot, FUN = mean)        # by year
aggregate(VOT ~ Consonant, data = df_vot, FUN = mean)   # by consonant

# Mean and SD by BOTH year and consonant
aggregate(VOT ~ year + Consonant, data = df_vot, FUN = mean)
aggregate(VOT ~ year + Consonant, data = df_vot, FUN = sd)
```

7e. Side-by-side histograms by group. To compare histograms fairly, give them the same x-axis limits (`xlim = c(0, 200)`) and y-axis limits (`ylim = c(0, 15)`). `par(mfrow = c(1, 2))` splits the plot window into 1 row and 2 columns.

```
# By year
par(mfrow = c(1, 2))
hist(df_vot$VOT[df_vot$year == 1971],
```

```
     xlim = c(0, 200), ylim = c(0, 15),
     main = "Cherokee VOT: 1971", xlab = "VOT (ms)",
     col = "lightblue", border = "navy")
hist(df_vot$VOT[df_vot$year == 2001],
     xlim = c(0, 200), ylim = c(0, 15),
     main = "Cherokee VOT: 2001", xlab = "VOT (ms)",
     col = "lightblue", border = "navy")

# By consonant
par(mfrow = c(1, 2))
hist(df_vot$VOT[df_vot$Consonant == "k"],
     xlim = c(0, 200), ylim = c(0, 15),
     main = "Cherokee VOT: /k/", xlab = "VOT (ms)",
     col = "lightblue", border = "navy")
hist(df_vot$VOT[df_vot$Consonant == "t"],
     xlim = c(0, 200), ylim = c(0, 15),
     main = "Cherokee VOT: /t/", xlab = "VOT (ms)",
     col = "lightblue", border = "navy")
```

All 4 combinations in a 2×2 grid (2 rows, 2 columns):

```
par(mfrow = c(2, 2), mar = c(4, 4, 2, 1))
hist(df_vot$VOT[df_vot$year == 1971 & df_vot$Consonant == "k"],
     xlim = c(0, 200), ylim = c(0, 10), main = "1971: /k/", xlab = "VOT (ms)",
     col = "lightblue", border = "navy")
hist(df_vot$VOT[df_vot$year == 1971 & df_vot$Consonant == "t"],
     xlim = c(0, 200), ylim = c(0, 10), main = "1971: /t/", xlab = "VOT (ms)",
     col = "lightblue", border = "navy")
hist(df_vot$VOT[df_vot$year == 2001 & df_vot$Consonant == "k"],
     xlim = c(0, 200), ylim = c(0, 10), main = "2001: /k/", xlab = "VOT (ms)",
     col = "lightblue", border = "navy")
hist(df_vot$VOT[df_vot$year == 2001 & df_vot$Consonant == "t"],
     xlim = c(0, 200), ylim = c(0, 10), main = "2001: /t/", xlab = "VOT (ms)",
     col = "lightblue", border = "navy")
```

Normal Q-Q plots for 1971 vs. 2001 side by side:

```
par(mfrow = c(1, 2))
qqnorm(df_vot$VOT[df_vot$year == 1971], main = "Q-Q Plot: 1971 VOT")
qqline(df_vot$VOT[df_vot$year == 1971], col = "red", lwd = 2)
qqnorm(df_vot$VOT[df_vot$year == 2001], main = "Q-Q Plot: 2001 VOT")
qqline(df_vot$VOT[df_vot$year == 2001], col = "red", lwd = 2)

par(mfrow = c(1, 1))   # ALWAYS reset the plot window to 1 row, 1 column
```

8. Practice questions

1. **Scale levels.** Run Section 3 again with `rating_levels <- 1:10` instead of `1:9`. How does the `rating_distribution` table change? Do the mean, median, or IQR in Section 5 change at all? Why or why not?

2. **Pooled vs. stratified data.** Compare the mean and SD of /k/ and /t/ in 1971 and 2001 (Section 7d). Why is it misleading to report only one overall mean VOT for the whole `cherokeeVOT.txt` dataset?

3. **Subsetting practice.** Write two lines of R code to draw a normal Q-Q plot (with a reference line) for /k/ in 2001 only. Does this subset look closer to a normal distribution than the pooled dataset?