Agentic AI Demo Report

Table of contents

```r
library(yaml)
library(readr)

metrics_path <- here::here("outputs","results","metrics.yml")
cleaning_path <- here::here("outputs","results","cleaning.yml")
fig_path <- here::here("outputs","figures","rt_hist.png")
freq_tidy_path <- here::here("outputs","results","character_frequency_model_tidy.csv")
freq_glance_path <- here::here("outputs","results","character_frequency_model_glance.csv")
freq_fig_path <- here::here("outputs","figures","character_frequency_rt_vs_freq.png")

metrics <- yaml::read_yaml(metrics_path)
cleaning <- yaml::read_yaml(cleaning_path)
freq_model_tidy <- read_csv(freq_tidy_path, show_col_types = FALSE)
freq_model_glance <- read_csv(freq_glance_path, show_col_types = FALSE)
N <- as.integer(metrics$n_obs)

# helpers for formatting
fmt3 <- function(x) sprintf("%.3f", x)
fmt6 <- function(x) sprintf("%.6f", x)

freq_coef <- freq_model_tidy$Estimate[freq_model_tidy$term == "log_freq"]
strokes_coef <- freq_model_tidy$Estimate[freq_model_tidy$term == "strokes"]
freq_pct <- (exp(freq_coef) - 1) * 100
strokes_pct <- (exp(strokes_coef) - 1) * 100
```

Overview

This report reads pre-computed outputs from the simple demo pipeline.

- Processed data: `outputs/data/processed.csv`
- Cleaning summary: `outputs/results/cleaning.yml`
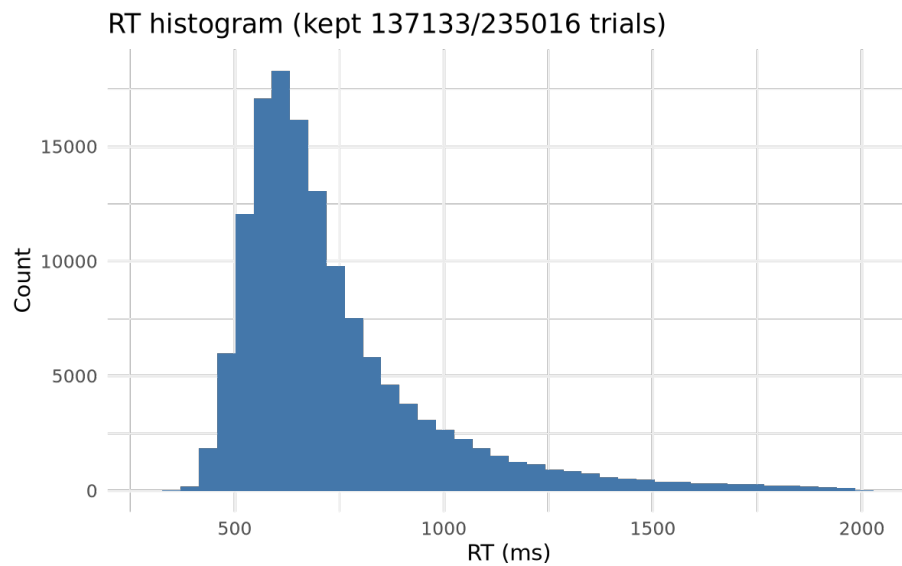- Model metrics: `outputs/results/metrics.yml`

Cleaning Summary

The pipeline kept 137133 of 235016 trials (dropped 97883). Settings: correct-only = TRUE, RT range = 200–2000 ms.

```r
data.frame(
  setting = c("correct_only","rt_min_ms","rt_max_ms","total_trials","kept_trials","dropped_t
  value = c(
    as.character(cleaning$trimming$correct_only),
    cleaning$trimming$rt_min_ms,
    cleaning$trimming$rt_max_ms,
    cleaning$counts$total_trials,
    cleaning$counts$kept_trials,
    cleaning$counts$dropped_trials
  )
)
```

```
        setting  value
1   correct_only   TRUE
2      rt_min_ms    200
3      rt_max_ms   2000
4   total_trials 235016
5    kept_trials 137133
6 dropped_trials  97883
```

RT Histogram (kept trials)

```r
knitr::include_graphics(fig_path)
```

RT histogram (kept 137133/235016 trials)

Model Metrics

Model: lm(mean_log_rt ~ log_freq + strokes) (N = 3852)

$R^2 = 0.434$.

```
cat(paste0("Adjusted R² = ", fmt3(as.numeric(metrics$adj_r2)), ".\n\n"))
```

Adjusted R² = 0.433.

```
cat(paste0("Residual sigma = ", fmt3(as.numeric(metrics$sigma)), ".\n\n"))
```

Residual sigma = 0.099.

```
cat("Information criteria:\n\n")
```

Information criteria:

```
print(data.frame(
  metric = c("AIC", "BIC"),
  value = c(fmt3(as.numeric(metrics$aic)), fmt3(as.numeric(metrics$bic)))
))
```

```
  metric     value
1    AIC -6851.160
2    BIC -6826.134
```

Coefficients:

```
data.frame(
  term = c("intercept","log_freq","strokes"),
```

```
  estimate = c(
    fmt6(as.numeric(metrics$coefficients$intercept)),
    fmt6(as.numeric(metrics$coefficients$log_freq)),
    fmt6(as.numeric(metrics$coefficients$strokes))
  )
)
```

```
       term   estimate
1 intercept   6.452355
2  log_freq  -0.070823
3   strokes   0.013355
```

Character Frequency Model

Character-level summaries (`outputs/data/character_frequency_model_data.csv`)
were modelled with median lexical decision times as the outcome and predictors
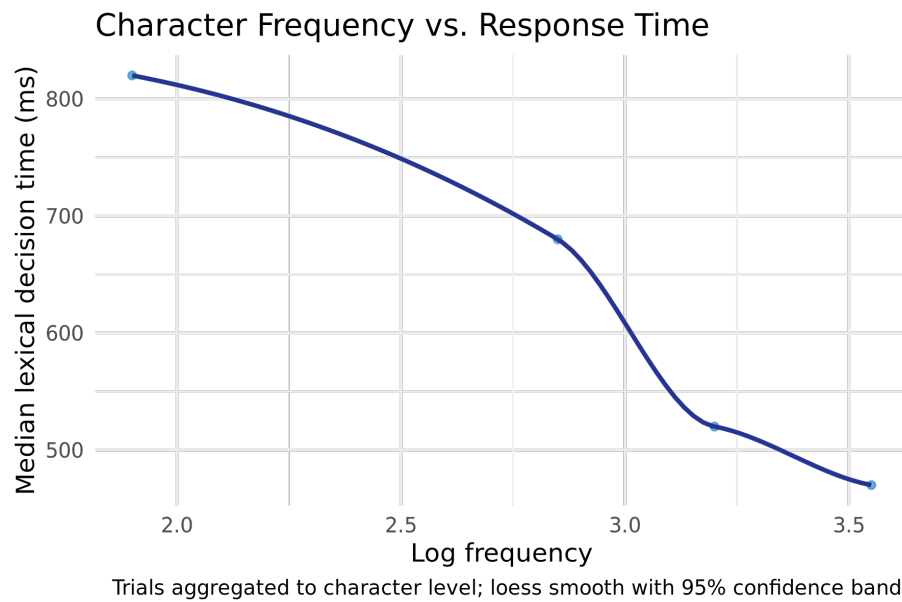`log_freq` and `strokes`.

```
knitr::kable(freq_model_tidy, digits = 3)
```

term	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	6.900	0.216	31.895	0.020
log_freq	-0.271	0.045	-6.063	0.104
strokes	0.032	0.012	2.612	0.233

95% uncertainty for the combined fit: $R^2 = 0.990$, sigma $= 0.043$.

- `log_freq`: -0.271 on the log scale $\rightarrow$ -23.737% faster median responses
  per one-unit increase in log frequency.
- `strokes`: 0.032 on the log scale $\rightarrow$ 3.263% slower median responses per
  additional stroke.

```
knitr::include_graphics(freq_fig_path)
```

Character Frequency vs. Response Time

Trials aggregated to character level; loess smooth with 95% confidence band.

Median response times fall sharply from rare to moderately frequent characters, but the loess curve flattens once log frequency exceeds roughly 3, suggesting diminishing speed gains for the most common characters. The widening confidence band at high frequency reflects the sparse sample, so the apparent plateau should be revisited when more characters are available, yet the current evidence aligns with classic frequency saturation once visual complexity is held constant.