

# Results

## Table of contents

Cleaning . . . . .	1
Baseline model: frequency and strokes . . . . .	2
Frequency effect with complexity adjustment . . . . .	3

```
library(yaml)
library(here)

here() starts at /root/repo

fmt3 <- function(x) sprintf("%.3f", x)
fmt6 <- function(x) sprintf("%.6f", x)
fmt1 <- function(x) sprintf("%.1f", x)

cleaning <- yaml::read_yaml(here("outputs", "results", "cleaning.yml"))
base <- yaml::read_yaml(here("outputs", "results", "base_lm.yml"))
freq_complexity <- yaml::read_yaml(here("outputs", "results", "freq_rt_complexity.yml"))
```

## Cleaning

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

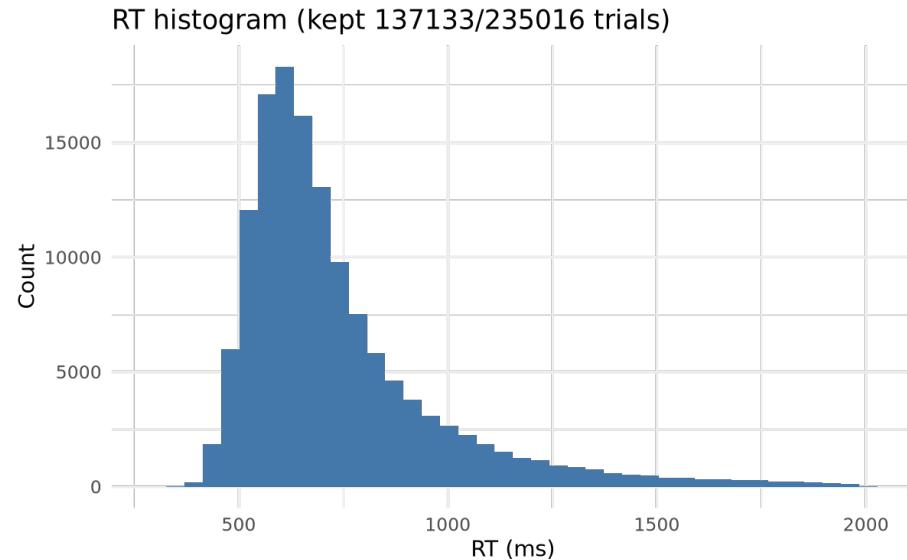
```
data.frame(
  setting = c(
    "correct_only",
    "rt_min_ms",
    "rt_max_ms",
    "total_trials",
    "kept_trials",
    "dropped_trials"
  ),
  value = c(
    as.character(cleaning$trimming$correct_only),
    cleaning$trimming$rt_min_ms,
    cleaning$trimming$rt_max_ms,
```

```

    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
knitr:::include_graphics(here("outputs", "figures", "rt_hist.png"))

```



### Baseline model: frequency and strokes

```

data.frame(
  term = c("intercept", "log_freq", "strokes"),
  estimate = c(
    fmt6(as.numeric(base$coefficients$intercept)),
    fmt6(as.numeric(base$coefficients$log_freq)),
    fmt6(as.numeric(base$coefficients$strokes))
  )
)

```

```

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

R^2 0.434; adjusted R^2 0.433; residual sigma 0.099. AIC -6851.160, BIC
-6826.134.

```

## Frequency effect with complexity adjustment

The GAM with a smooth log-frequency term and linear stroke adjustment used 3852 characters. Moving from the 10th to 50th percentile in frequency speeds responses by about 82.1 ms, whereas gains shrink to 38.0 ms between the 50th and 90th percentile, indicating a flattening benefit at the top end.

```

quantile_rows <- freq_complexity$quantile_estimates
data.frame(
  quantile = names(quantile_rows),
  log_freq = vapply(quantile_rows, function(x) x$log_freq, numeric(1)),
  freq = vapply(quantile_rows, function(x) x$freq, numeric(1)),
  rt_ms = vapply(quantile_rows, function(x) x$rt_ms, numeric(1)),
  rt_ms_lower = vapply(quantile_rows, function(x) x$rt_ms_lower, numeric(1)),
  rt_ms_upper = vapply(quantile_rows, function(x) x$rt_ms_upper, numeric(1))
)

quantile log_freq    freq  rt_ms rt_ms_lower rt_ms_upper
10th      10th    0.0310 1.0315  752.62      748.68     756.58
50th      50th    0.6051 1.8315  670.56      667.11     674.03
90th      90th    2.1035 8.1949  632.59      628.04     637.18

```

```

knitr::include_graphics(here("outputs", "figures", "freq_rt_vs_frequency.png"))

```

Faster responses for higher-frequency characters  
Character-level median RTs with GAM smooth adjusted for strok

