

Data481_Superbowl

jph

2026-01-15

```
library(tidyverse)
```

```
## Warning: package 'ggplot2' was built under R version 4.3.3
```

```
## Warning: package 'purrr' was built under R version 4.3.3
```

```
## Warning: package 'lubridate' was built under R version 4.3.3
```

```
## — Attaching core tidyverse packages — tidyverse 2.0.0 —
## ✓ dplyr      1.1.4      ✓ readr      2.1.5
## ✓ forcats    1.0.0      ✓ stringr    1.5.1
## ✓ ggplot2    3.5.2      ✓ tibble     3.2.1
## ✓ lubridate  1.9.4      ✓ tidyr      1.3.1
## ✓ purrr      1.0.4
## — Conflicts — tidyverse_conflicts() —
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
youtube <- read_csv("~/Downloads/youtube.csv")
```

```
## Rows: 247 Columns: 25
## — Column specification —
## Delimiter: ","
## chr  (10): brand, superbowl_ads_dot_com_url, youtube_url, id, kind, etag, ti...
## dbl  (7): year, view_count, like_count, dislike_count, favorite_count, comm...
## lgl  (7): funny, show_product_quickly, patriotic, celebrity, danger, animal...
## dtm  (1): published_at
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
head(youtube)
```

```
## # A tibble: 6 × 25
##   year brand      superbowl_ads_dot_com...1 youtube_url funny show_product_quickly
##   <dbl> <chr>      <chr>                                <chr>      <lgl> <lgl>
## 1  2018 Toyota    https://superbowl-ads.... https://ww... FALSE FALSE
## 2  2020 Bud Light https://superbowl-ads.... https://ww... TRUE  TRUE
## 3  2006 Bud Light https://superbowl-ads.... https://ww... TRUE  FALSE
## 4  2018 Hynudai   https://superbowl-ads.... https://ww... FALSE TRUE
## 5  2003 Bud Light https://superbowl-ads.... https://ww... TRUE  TRUE
## 6  2020 Toyota    https://superbowl-ads.... https://ww... TRUE  TRUE
## # i abbreviated name: 1superbowl_ads_dot_com_url
## # i 19 more variables: patriotic <lgl>, celebrity <lgl>, danger <lgl>,
## #   animals <lgl>, use_sex <lgl>, id <chr>, kind <chr>, etag <chr>,
## #   view_count <dbl>, like_count <dbl>, dislike_count <dbl>,
## #   favorite_count <dbl>, comment_count <dbl>, published_at <dtm>,
## #   title <chr>, description <chr>, thumbnail <chr>, channel_title <chr>,
## #   category_id <dbl>
```

```
library(tidyverse)

ad_types <- c("funny", "show_product_quickly", "patriotic",
             "celebrity", "danger", "animals", "use_sex")

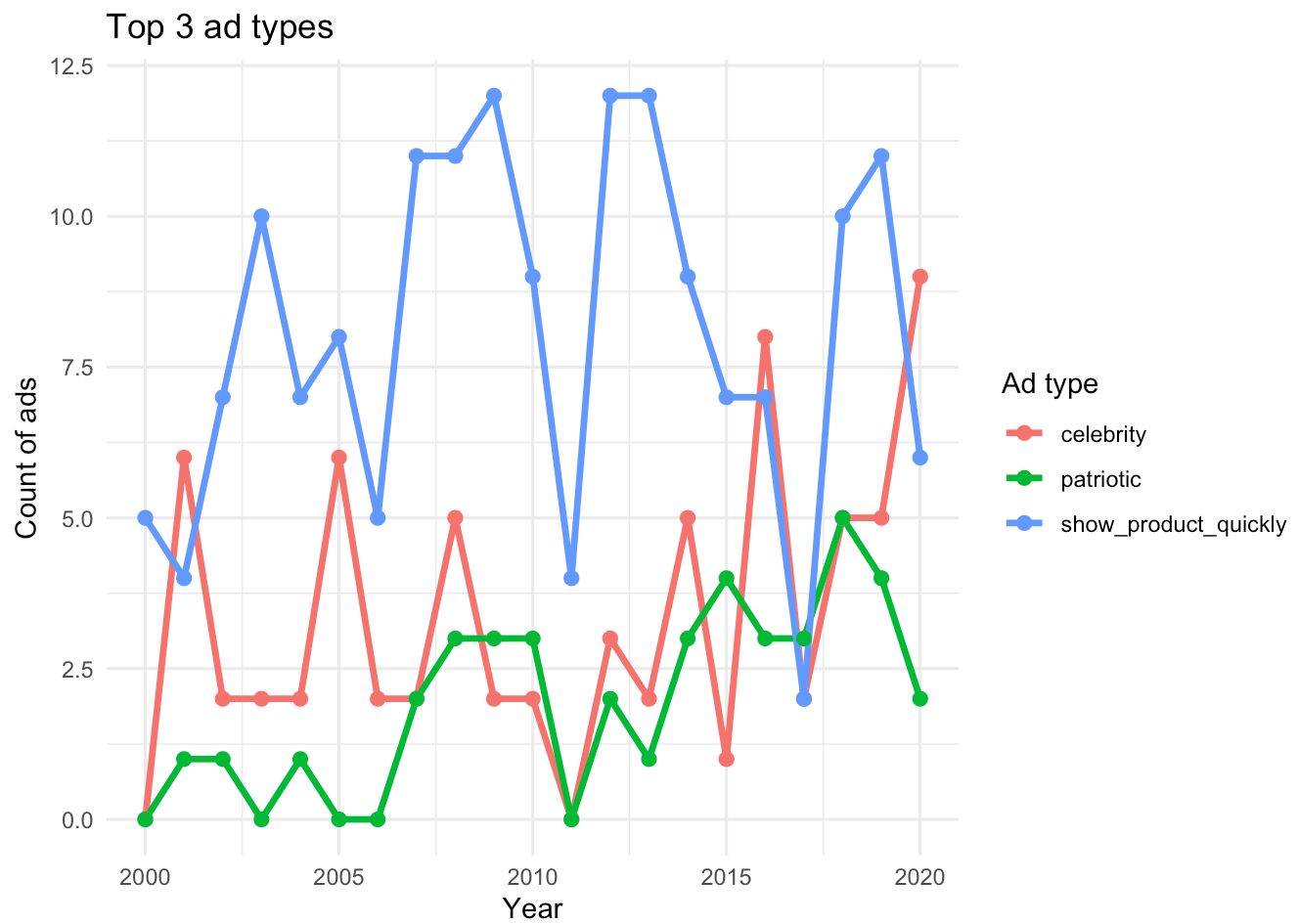
counts_by_year <- youtube %>%
  select(year, all_of(ad_types)) %>%
  pivot_longer(cols = all_of(ad_types),
               names_to = "ad_type", values_to = "used") %>%
  group_by(year, ad_type) %>%
  summarise(count_used = sum(used, na.rm = TRUE), .groups = "drop")

trend_info <- counts_by_year %>%
  group_by(ad_type) %>%
  summarise(
    slope = coef(lm(count_used ~ year))[2],
    total = sum(count_used),
    .groups = "drop"
  )

top_up <- trend_info %>% arrange(desc(slope)) %>% slice(1:3) %>% pull(ad_type)
top_down <- trend_info %>% arrange(slope) %>% slice(1:3) %>% pull(ad_type)
top_total <- trend_info %>% arrange(desc(total)) %>% slice(1:3) %>% pull(ad_type)

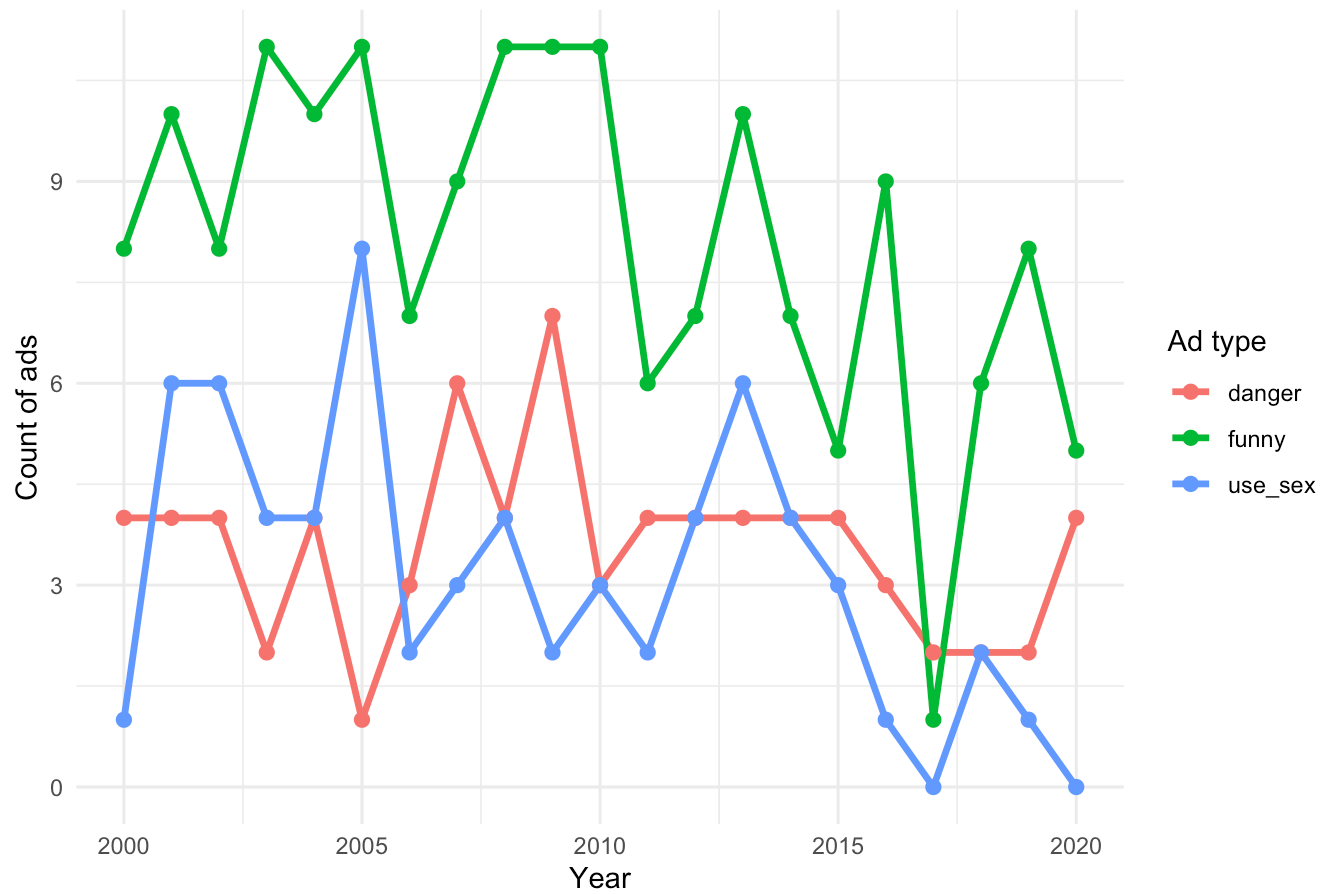
plot_3 <- function(types, title_text) {
  counts_by_year %>%
    filter(ad_type %in% types) %>%
    ggplot(aes(x = year, y = count_used, color = ad_type)) +
    geom_line(linewidth = 1.2) +
    geom_point(size = 2.2) +
    labs(title = title_text, x = "Year", y = "Count of ads", color = "Ad type") +
    theme_minimal()
}

plot_3(top_up, "Top 3 ad types")
```

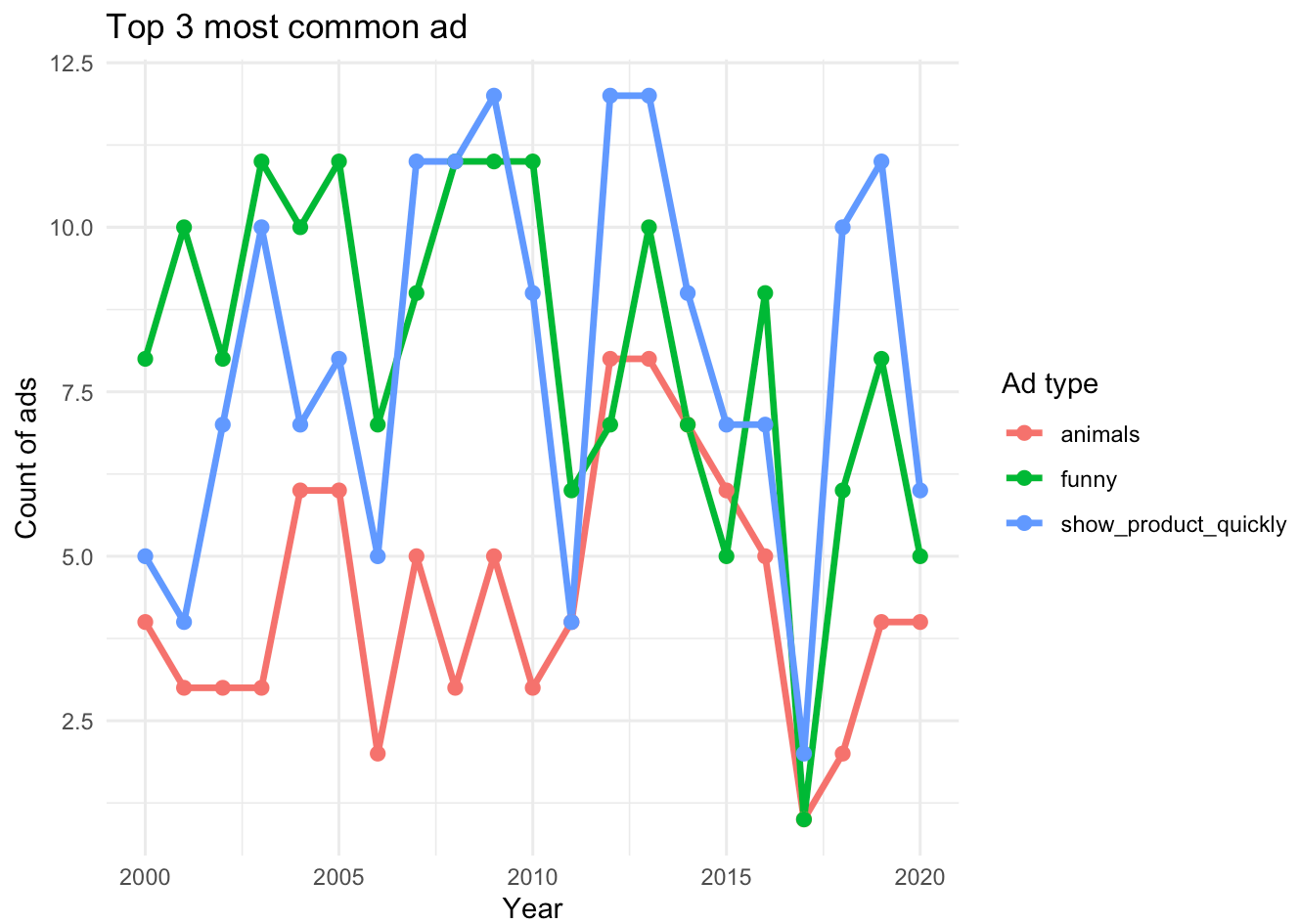


```
plot_3(top_down, "Top 3 ad types ")
```

Top 3 ad types



```
plot_3(top_total, "Top 3 most common ad")
```



```
library(tidyverse)

ad_types <- c("funny", "show_product_quickly", "patriotic",
             "celebrity", "danger", "animals", "use_sex")

counts_by_year <- youtube %>%
  select(year, all_of(ad_types)) %>%
  pivot_longer(
    cols = all_of(ad_types),
    names_to = "ad_type",
    values_to = "used"
  ) %>%
  group_by(year, ad_type) %>%
  summarise(count_used = sum(used, na.rm = TRUE), .groups = "drop")

yearly_change <- counts_by_year %>%
  arrange(ad_type, year) %>%
  group_by(ad_type) %>%
  mutate(delta = count_used - lag(count_used, 1)) %>%
  ungroup() %>%
  filter(!is.na(delta))

change_wide_1 <- yearly_change %>%
  select(year, ad_type, delta) %>%
  pivot_wider(names_from = ad_type, values_from = delta)

change_wide_2 <- yearly_change %>%
  select(year, ad_type, delta) %>%
  pivot_wider(names_from = ad_type, values_from = delta)

lm_funny <- lm(
  funny ~ celebrity + use_sex + animals + patriotic + danger + show_product_quickly,
  data = change_wide_1
)
cat("MODEL 1: ΔFunny ~ predictors\n")
```

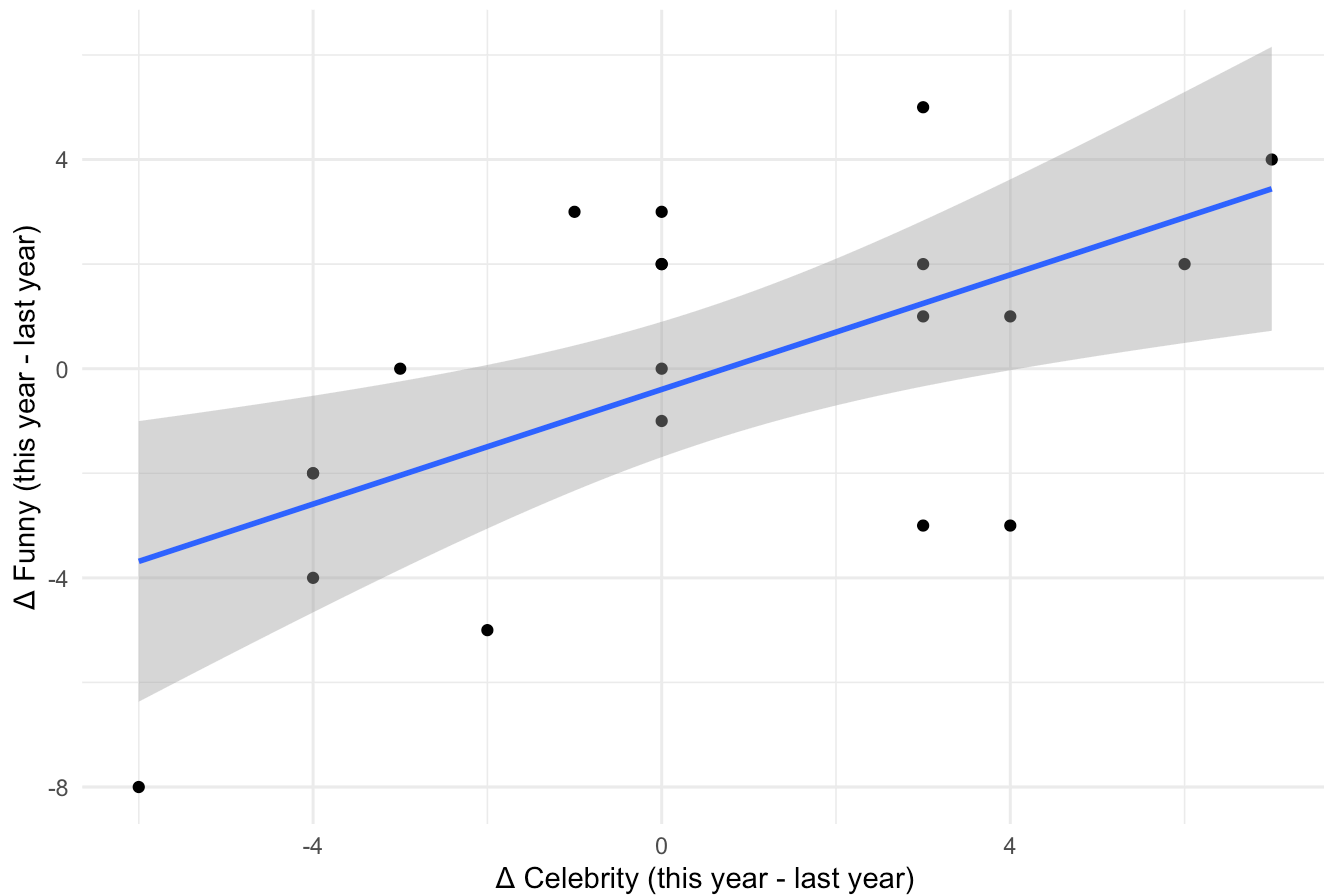
```
## MODEL 1: ΔFunny ~ predictors
```

```
print(summary(lm_funny))
```

```
##
## Call:
## lm(formula = funny ~ celebrity + use_sex + animals + patriotic +
##     danger + show_product_quickly, data = change_wide_1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.6151 -1.6904  0.5798  1.2950  3.3965
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.325904    0.495637  -0.658   0.5223
## celebrity       0.402937    0.159205   2.531   0.0251 *
## use_sex        0.003895    0.274345   0.014   0.9889
## animals        0.004189    0.368834   0.011   0.9911
## patriotic     -0.324530    0.433954  -0.748   0.4679
## danger        -0.172657    0.374100  -0.462   0.6520
## show_product_quickly 0.544592    0.186783   2.916   0.0120 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.182 on 13 degrees of freedom
## Multiple R-squared:  0.7032, Adjusted R-squared:  0.5663
## F-statistic: 5.134 on 6 and 13 DF,  p-value: 0.006559
```

```
ggplot(change_wide_1, aes(x = celebrity, y = funny)) +
  geom_point() +
  geom_smooth(method = "lm", se = TRUE) +
  labs(
    title = "Yearly change relationship: ΔFunny vs ΔCelebrity",
    x = "Δ Celebrity (this year - last year)",
    y = "Δ Funny (this year - last year)"
  ) +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

Yearly change relationship: Δ Funny vs Δ Celebrity

```
lm_spq <- lm(
  show_product_quickly ~ celebrity + use_sex + animals + patriotic + danger + funny,
  data = change_wide_2
)

ggplot(change_wide_2, aes(x = funny, y = show_product_quickly)) +
  geom_point() +
  geom_smooth(method = "lm", se = TRUE) +
  labs(
    title = "Yearly change relationship:  $\Delta$ Show Product Quickly vs  $\Delta$ Funny",
    x = " $\Delta$  Funny (this year - last year)",
    y = " $\Delta$  Show Product Quickly (this year - last year)"
  ) +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

Yearly change relationship: Δ Show Product Quickly vs Δ Funny