Informal Rust Gamedev in 2024 Survey

Jan Hohenheim

2024-05-04

Dependencies

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(glue)
library(ggthemes)
library(latex2exp)

theme_set(theme_solarized_2())
```

Data Cleaning

See https://www.reddit.com/r/rust_gamedev/comments/1cka6n8/informal_rust_gamedev_in_2024_survey_results/

```
dat.raw <- read_csv("data_original.csv")
```

```
## Rows: 410 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (4): Timestamp, How are you currently using Rust to make games? Select ...
## dbl (13): What are the biggest barriers to your success when making games in...
##
## 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.
```

```
dat <- dat.raw |>
  # Change names
  ## Timestamp
  rename_at(1, ~ "timestamp") |>
  ## How are you currently using Rust to make games?
  rename_at(2, ~ "usage") |>
  ## What Rust-based game engine do you primarily use?
```

```
  rename_at(3, ~ "engine") |>
  ## Long compile and iteration times
  rename_at(4, ~ "bad_iteration_time") |>
  ## Problems with Rust itself (other than compile times)
  rename_at(5, ~ "bad_rust") |>
  ## Problems in platform-abstracting crates like winit or wgpu
  rename_at(6, ~ "bad_abstraction") |>
  ## Inadequate learning materials or docs
  rename_at(7, ~ "bad_docs") |>
  ## Poor tooling for artists and game designers
  rename_at(8, ~ "bad_tooling") |>
  ## Difficulty paying to get open source problems fixed
  rename_at(9, ~ "bad_paying_for_bugs") |>
  ## Lack of console support
  rename_at(10, ~ "bad_console") |>
  ## Immature mobile support
  rename_at(11, ~ "bad_mobile") |>
  ## Immature web support
  rename_at(12, ~ "bad_web") |>
  ## Bugs in the engine I use
  rename_at(13, ~ "bad_engine_bugs") |>
  ## Missing features in the engine I use
  rename_at(14, ~ "bad_engine_features") |>
  ## Difficulty hiring experts who know Rust
  rename_at(15, ~ "bad_hiring") |>
  ## Poor performance
  rename_at(16, ~ "bad_performance") |>
  ## If you could magically add or fix three things about Rust itself, what would they be?
  rename_at(17, ~ "magic_fix") |>
  # Rename factors
  mutate(usage = fct_recode(usage,
    gamedev_serious_hobby = "I have at least one serious hobbyist project that I have or am planning to
    tooling_creator = "Actually, I only use Rust to make game engines or tools for gamedev.",
    gamedev_commercial_solo = "I work by myself, but have a project that I have or am planning to sell.
    gamedev_financial_support = "I work by myself or in a tiny team, and am attempting to support mysel
    tooling_company = "I am part of a company that uses Rust game tools to make things that are not game
    gamedev_learner = "I'm still learning.",
  )) |>
  # Classify custom answers. There's probably a better way to do this, sorry
  mutate(usage = usage |>
    fct_recode(gamedev_serious_hobby = "I've used Rust for making tools while working in the games indus
    fct_recode(gamedev_serious_hobby = "I am part of a large open source game written in Rust") |>

    fct_recode(gamedev_commercial_solo = "I have a game engine and game editor in rust that I am looking

    fct_recode(gamedev_financial_support = "Both make games and tools using Rust in a tiny team to suppo
    fct_recode(gamedev_financial_support = "I am part of a company that is using Rust to make games.")

    fct_recode(tooling_company = "I work for Foresight, making CAD tools using bevy and rust.") |>
    fct_recode(tooling_company = "I use Rust game tools for academic research.") |>

    fct_recode(gamedev_learner = "I write rust code but not much game-dev but I dabbled with bevy a lit
```

```
    fct_recode(gamedev_casual_hobby = "Hobbyist game developer") |>
    fct_recode(gamedev_casual_hobby = "\"I have at least one serious hobbyist project\" … for which \"sl
    fct_recode(gamedev_casual_hobby = "Only as a hobby") |>
    fct_recode(gamedev_casual_hobby = "I am still in university, so not working commercially, but use Ru
    fct_recode(gamedev_casual_hobby = "I use rust to make games primarily as a hobby, but technically ha
    fct_recode(gamedev_casual_hobby = "I use Bevy to create games and teach my son about developing and
    fct_recode(gamedev_casual_hobby = "I sometimes use Rust/Bevy for Game Jam entries as a change from C
    fct_recode(gamedev_casual_hobby = "I make games for my kids") |>

    fct_recode(gamedev_quit = "I used to make games in Rust.") |>
    fct_recode(gamedev_quit = "Recently moved game project away from Rust") |>
    fct_recode(gamedev_quit = "im not using rust for game dev anymore - turns out it sucks ass for it")
    fct_recode(gamedev_quit = "Tried Rust for gamedev. Too much ceremony needed for everything. It was n
    fct_recode(gamedev_quit = "I was working by myself to financially support with games made in Rust, l
    fct_recode(gamedev_quit = "I have made games in Rust in the past") |>

    fct_recode(other = "Paid contractor making a metaverse client") |>
    fct_recode(other = "I worked for a games company before") |>
    fct_recode(other = "I work with Rust outside of games but I want to make games in it when the tools
    fct_recode(other = "I was \"I  have at least one serious hobbyist project that I have or am planning
    fct_recode(other = "I use Rust game tools to make things that are not games as a hobby or passion.")
    fct_recode(other = "I freelance for companies shipping AAA games.") |>
    fct_recode(other = "I don't make games in Rust.")
) |>
  mutate(engine = factor(engine) |>
    fct_recode(fyrox = "Fyrox") |>
    fct_recode(fyrox = "Been experimenting, but mostly fyrox.") |>
    fct_recode(fyrox = "Tried Fyrox. Great engine but Rust was the limitation.") |>
    fct_recode(bevy = "Bevy") |>
    fct_recode(bevy_extra = "Bevy + In House Engine") |>
    fct_recode(bevy_extra = "both Bevy & no-engine / in-house, depending on client and project") |>
    fct_recode(bevy_extra = "In house and Bevy") |>
    fct_recode(chuot = "Chuột") |>
    fct_recode(comfy = "Comfy") |>
    fct_recode(custom = "Custom-written wgpu-based 2D engine (for Visual Novels)") |>
    fct_recode(custom = "No engine / in-house engine") |>
    fct_recode(custom = "My custom engine") |>
    fct_recode(gamercade = "Gamercade") |>
    fct_recode(godot = "gdnative (Rust and Godot)") |>
    fct_recode(godot = "Godot") |>
    fct_recode(godot = "Godot with rust bindings (gdext)") |>
    fct_recode(godot = "Godot-GDNative") |>
    fct_recode(godot = "godot-rust") |>
    fct_recode(godot = "Godot-Rust") |>
    fct_recode(godot = "Godot + gdnative") |>
    fct_recode(godot = "Godot/gdext") |>
    fct_recode(raylib = "Raylib Rust bindings") |>
    fct_recode(raylib = "Raylib-ffi") |>
    fct_recode(none = "no engine, but we use specs + wgpu + conrod/iced for gui") |>
    fct_recode(none = "No engine with wgpu, bevy ecs, egui.") |>
    fct_recode(none = "No engine") |>
    fct_recode(none = "wgpu") |>
    fct_recode(none = "SDL") |>
```

```
    fct_recode(none = "Rend3/WGPU") |>
    fct_recode(none = "None") |>
    fct_recode(tetra = "Tetra") |>
    fct_recode(speedy2d = "Speedy2D") |>
    fct_recode(quad = "Macroquad") |>
    fct_recode(quad = "miniquad") |>
    fct_recode(piston = "Piston") |>
    fct_recode(ggez = "Good Web Game") |>
    fct_recode(other = "I have only tried Fyrox and Bevy but both are currently lacking")
  )

dat |> summary()
```

```
##   timestamp                                usage         engine
##  Length:410         gamedev_serious_hobby    :127   bevy   :289
##  Class :character   gamedev_learner          : 83   custom : 63
##  Mode  :character   gamedev_commercial_solo  : 51   quad   : 15
##                     gamedev_financial_support: 44   godot  :  9
##                     tooling_creator          : 36   none   :  7
##                     other                    : 30   ggez   :  6
##                     (Other)                  : 39   (Other): 21
##  bad_iteration_time    bad_rust    bad_abstraction    bad_docs
##  Min.   :0.000      Min.   :0.00   Min.   :0.000   Min.   :0.000
##  1st Qu.:1.000      1st Qu.:0.00   1st Qu.:0.000   1st Qu.:1.000
##  Median :2.000      Median :1.00   Median :1.000   Median :2.000
##  Mean   :2.473      Mean   :1.32   Mean   :1.383   Mean   :2.012
##  3rd Qu.:4.000      3rd Qu.:2.00   3rd Qu.:2.000   3rd Qu.:3.000
##  Max.   :5.000      Max.   :5.00   Max.   :5.000   Max.   :5.000
##
##   bad_tooling    bad_paying_for_bugs  bad_console      bad_mobile
##  Min.   :0.000   Min.   :0.0000      Min.   :0.000   Min.   :0.000
##  1st Qu.:1.000   1st Qu.:0.0000      1st Qu.:0.000   1st Qu.:0.000
##  Median :2.000   Median :0.0000      Median :0.000   Median :0.000
##  Mean   :2.307   Mean   :0.5488      Mean   :1.117   Mean   :1.278
##  3rd Qu.:4.000   3rd Qu.:1.0000      3rd Qu.:2.000   3rd Qu.:2.000
##  Max.   :5.000   Max.   :5.0000      Max.   :5.000   Max.   :5.000
##
##     bad_web      bad_engine_bugs  bad_engine_features   bad_hiring
##  Min.   :0.000   Min.   :0.0000   Min.   :0.000       Min.   :0.0000
##  1st Qu.:0.000   1st Qu.:0.0000   1st Qu.:1.000       1st Qu.:0.0000
##  Median :0.000   Median :1.0000   Median :3.000       Median :0.0000
##  Mean   :1.102   Mean   :0.9146   Mean   :2.478       Mean   :0.6512
##  3rd Qu.:2.000   3rd Qu.:1.0000   3rd Qu.:4.000       3rd Qu.:1.0000
##  Max.   :5.000   Max.   :5.0000   Max.   :5.000       Max.   :5.0000
##
##  bad_performance   magic_fix
##  Min.   :0.0000   Length:410
##  1st Qu.:0.0000   Class :character
##  Median :0.0000   Mode  :character
##  Mean   :0.5854
##  3rd Qu.:1.0000
##  Max.   :5.0000
##
```

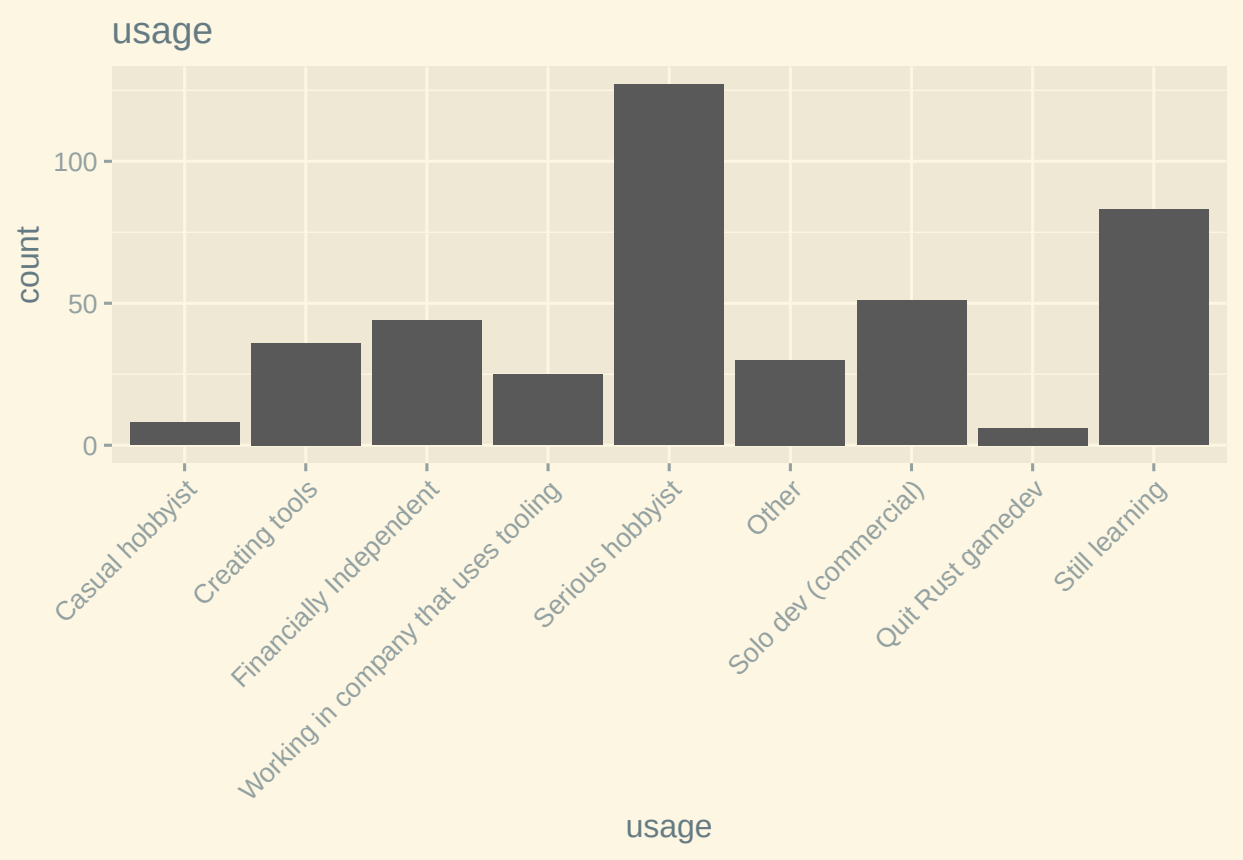
```r
dat |> head()
```

```
## # A tibble: 6 x 17
##   timestamp    usage engine bad_iteration_time bad_rust bad_abstraction bad_docs
##   <chr>        <fct> <fct>               <dbl>    <dbl>           <dbl>    <dbl>
## 1 4/29/2024 1~ game~ bevy                    0        2               2        0
## 2 4/29/2024 1~ game~ bevy                    1        0               0        1
## 3 4/29/2024 1~ game~ bevy                    4        2               2        1
## 4 4/29/2024 1~ game~ none                    0        0               1        0
## 5 4/29/2024 1~ game~ bevy                    0        0               0        1
## 6 4/29/2024 1~ game~ bevy                    0        1               3        4
## # i 10 more variables: bad_tooling <dbl>, bad_paying_for_bugs <dbl>,
## #   bad_console <dbl>, bad_mobile <dbl>, bad_web <dbl>, bad_engine_bugs <dbl>,
## #   bad_engine_features <dbl>, bad_hiring <dbl>, bad_performance <dbl>,
## #   magic_fix <chr>
```

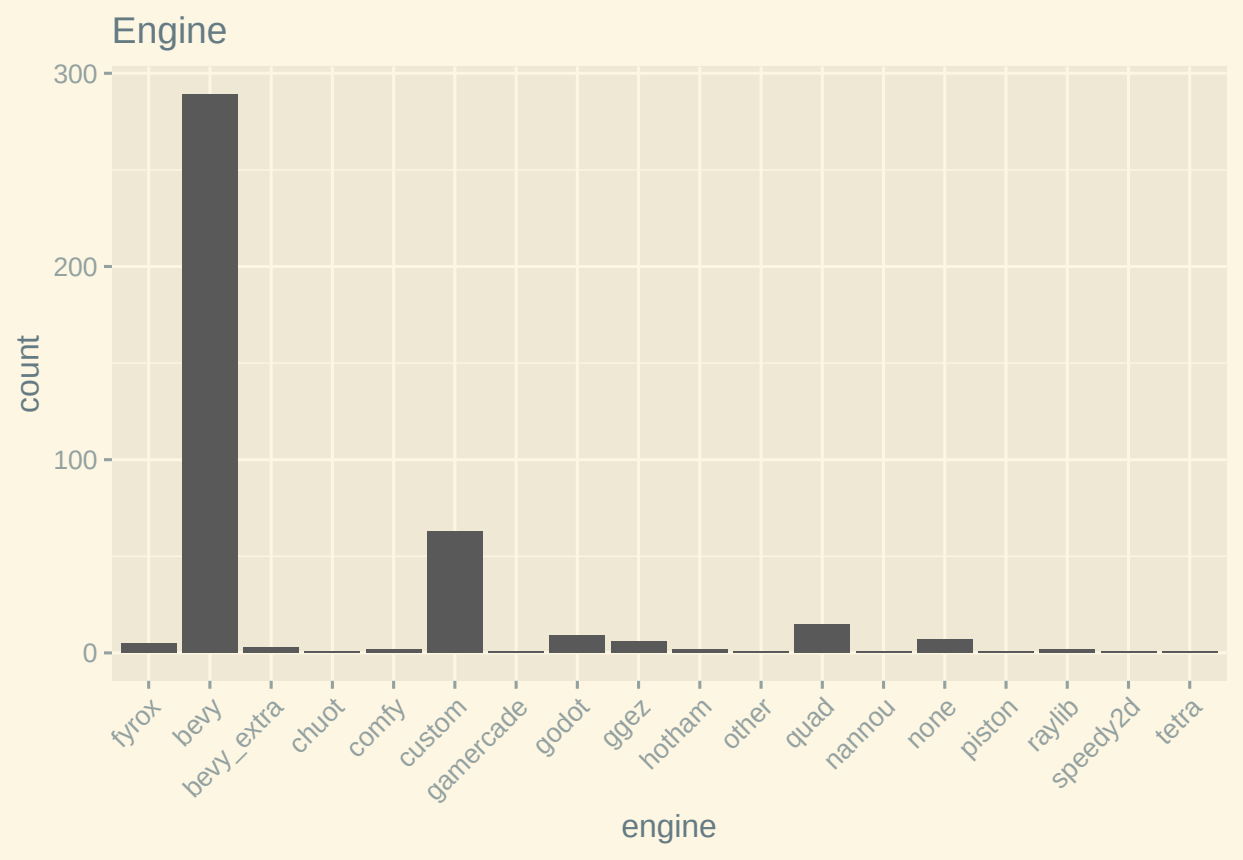
```r
dat |> write_csv("data_cleaned.csv")
```

Generic Plots

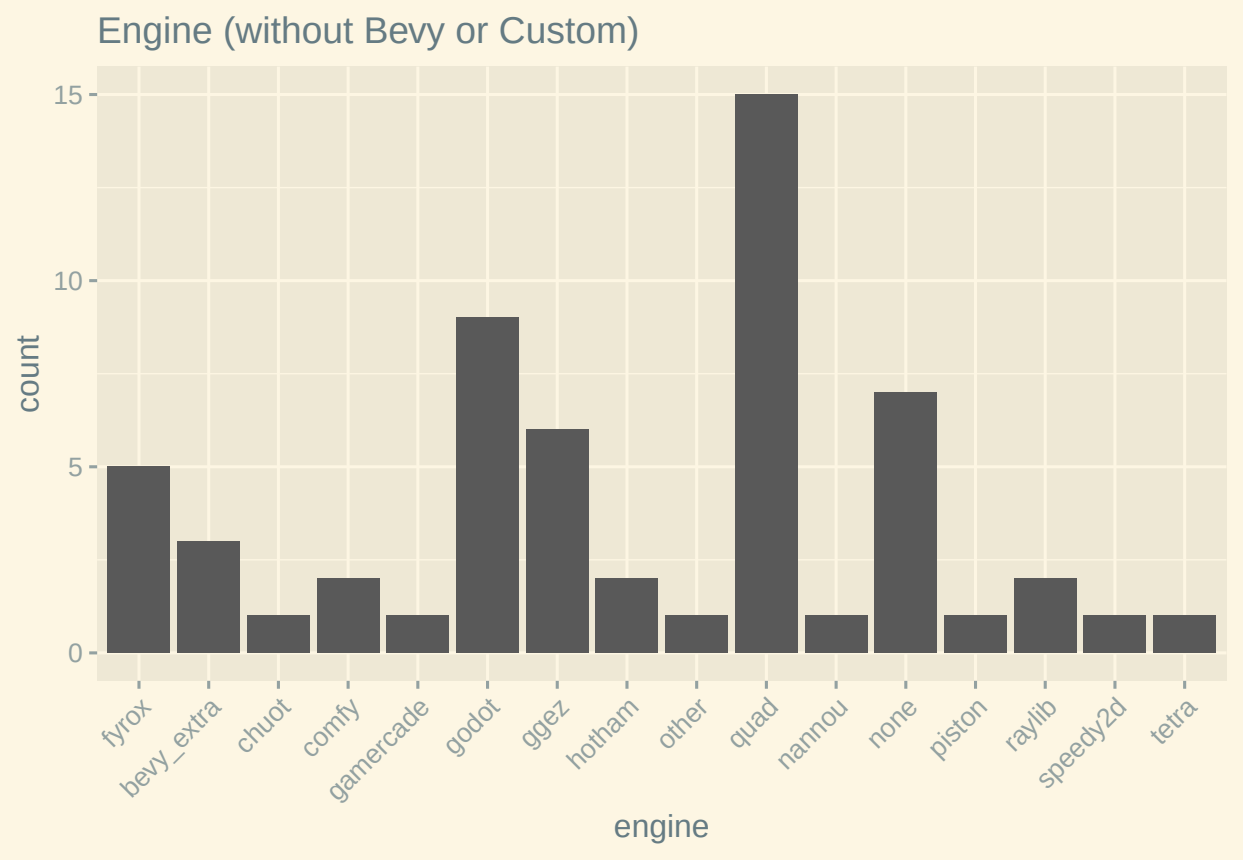
```r
dat |>
  ggplot(aes(x = usage)) +
  geom_bar() +
  ggtitle("usage") +
  # use abbreviated labels
  scale_x_discrete(labels = c(
    "gamedev_serious_hobby" = "Serious hobbyist",
    "tooling_creator" = "Creating tools",
    "gamedev_commercial_solo" = "Solo dev (commercial)",
    "gamedev_financial_support" = "Financially Independent",
    "tooling_company" = "Working in company that uses tooling",
    "gamedev_learner" = "Still learning",
    "gamedev_casual_hobby" = "Casual hobbyist",
    "gamedev_quit" = "Quit Rust gamedev",
    "other" = "Other"
  )) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



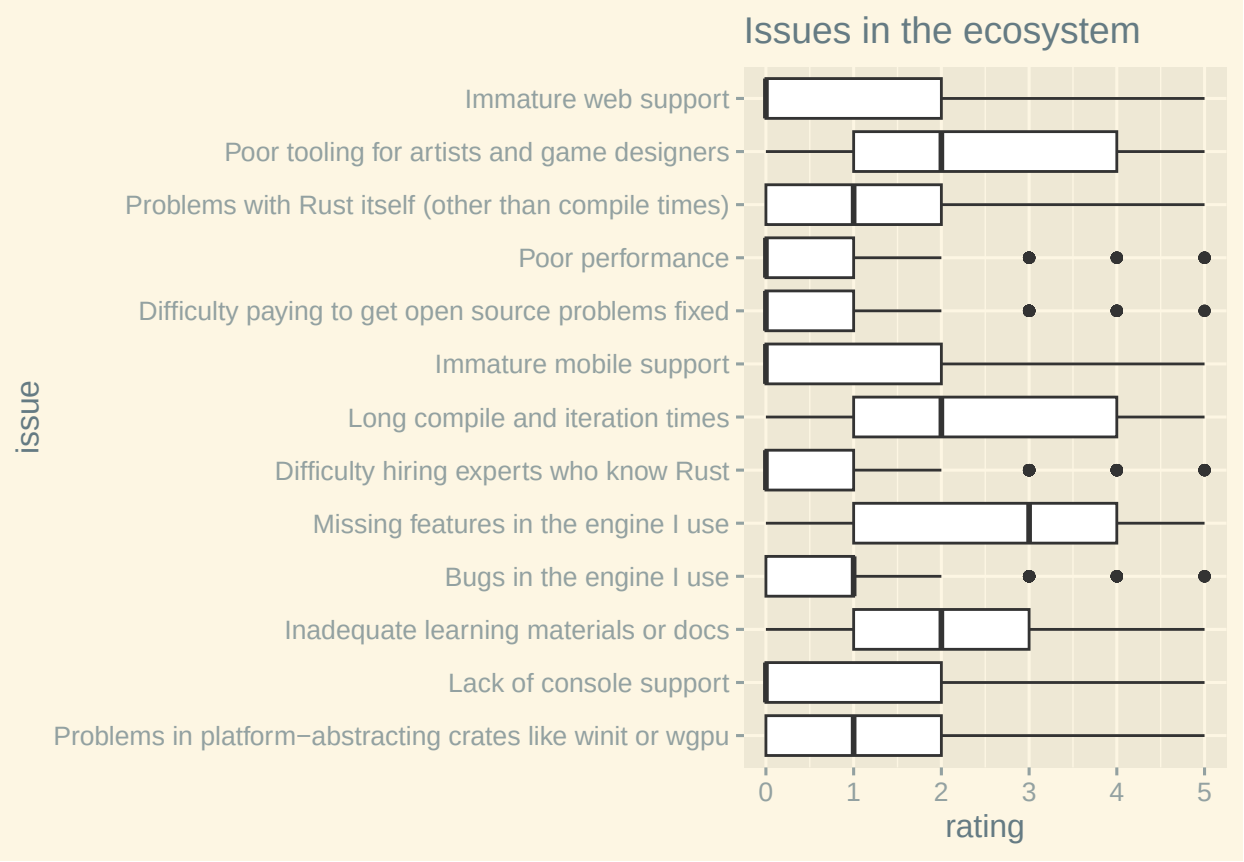

```
dat |>
  ggplot(aes(x = engine)) +
  geom_bar() +
  ggtitle("Engine") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```
dat |>
  filter(engine != "bevy") |>
  filter(engine != "custom") |>
  ggplot(aes(x = engine)) +
  geom_bar() +
  ggtitle("Engine (without Bevy or Custom)") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```
dat |>
  pivot_longer(cols = 4:16, names_to = "issue", values_to = "rating") |>
  ggplot(aes(x = issue, y = rating)) +
  geom_boxplot() +
  ggtitle("Issues in the ecosystem") +
  scale_x_discrete(labels = c(
    "bad_iteration_time" = "Long compile and iteration times",
    "bad_rust" = "Problems with Rust itself (other than compile times)",
    "bad_abstraction" = "Problems in platform-abstracting crates like winit or wgpu",
    "bad_docs" = "Inadequate learning materials or docs",
    "bad_tooling" = "Poor tooling for artists and game designers",
    "bad_paying_for_bugs" = "Difficulty paying to get open source problems fixed",
    "bad_console" = "Lack of console support",
    "bad_mobile" = "Immature mobile support",
    "bad_web" = "Immature web support",
    "bad_engine_bugs" = "Bugs in the engine I use",
    "bad_engine_features" = "Missing features in the engine I use",
    "bad_hiring" = "Difficulty hiring experts who know Rust",
    "bad_performance" = "Poor performance"
  )) +
  coord_flip()
```



Analysis

```
# TODO: change this, this is horrible.
lm(bad_iteration_time ~ engine, data = dat) |> summary()
```

```
##
## Call:
## lm(formula = bad_iteration_time ~ engine, data = dat)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -2.574 -1.574  0.000  1.426  3.000
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)         1.4000     0.7424   1.886   0.0601 .
## enginebevy          1.1744     0.7488   1.568   0.1176
## enginebevy_extra    0.9333     1.2124   0.770   0.4419
## enginechuot         2.6000     1.8186   1.430   0.1536
## enginecomfy         1.1000     1.3890   0.792   0.4289
## enginecustom        0.9016     0.7713   1.169   0.2432
## enginegamercade    -0.4000     1.8186  -0.220   0.8260
## enginegodot         0.4889     0.9260   0.528   0.5978
## engineggez          0.4333     1.0052   0.431   0.6667
## enginehotham        0.1000     1.3890   0.072   0.9426
## engineother         3.6000     1.8186   1.980   0.0484 *
```

```
## enginequad         1.2667     0.8573   1.478   0.1403
## enginenannou      -0.4000     1.8186  -0.220   0.8260
## enginenone         0.6000     0.9721   0.617   0.5374
## enginepiston      -1.4000     1.8186  -0.770   0.4419
## engineraylib       3.1000     1.3890   2.232   0.0262 *
## enginespeedy2d    -1.4000     1.8186  -0.770   0.4419
## enginetetra       -0.4000     1.8186  -0.220   0.8260
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.66 on 392 degrees of freedom
## Multiple R-squared:  0.04917,    Adjusted R-squared:  0.007931
## F-statistic: 1.192 on 17 and 392 DF,  p-value: 0.2672
```

```
lm(bad_rust ~ engine, data = dat) |> summary()
```

```
##
## Call:
## lm(formula = bad_rust ~ engine, data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.8000 -1.1765 -0.1765  0.8235  3.8235
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)         1.6000     0.6593   2.427   0.0157 *
## enginebevy         -0.4235     0.6650  -0.637   0.5246
## enginebevy_extra    0.7333     1.0767   0.681   0.4962
## enginechuot        -0.6000     1.6151  -0.372   0.7105
## enginecomfy         1.9000     1.2335   1.540   0.1243
## enginecustom        0.1460     0.6850   0.213   0.8313
## enginegamercade    -1.6000     1.6151  -0.991   0.3225
## enginegodot         0.1778     0.8224   0.216   0.8290
## engineggez         -0.2667     0.8928  -0.299   0.7653
## enginehotham       -0.6000     1.2335  -0.486   0.6269
## engineother         1.4000     1.6151   0.867   0.3866
## enginequad          0.2000     0.7614   0.263   0.7929
## enginenannou       -1.6000     1.6151  -0.991   0.3225
## enginenone         -0.6000     0.8633  -0.695   0.4875
## enginepiston       -1.6000     1.6151  -0.991   0.3225
## engineraylib        0.9000     1.2335   0.730   0.4661
## enginespeedy2d     -1.6000     1.6151  -0.991   0.3225
## enginetetra        -1.6000     1.6151  -0.991   0.3225
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.474 on 392 degrees of freedom
## Multiple R-squared:  0.05653,    Adjusted R-squared:  0.01561
## F-statistic: 1.382 on 17 and 392 DF,  p-value: 0.1415
```

```
lm(bad_abstraction ~ engine, data = dat) |> summary()
```

```
##
## Call:
```

```
## lm(formula = bad_abstraction ~ engine, data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.0000 -1.2111 -0.2111  0.7889  3.7889
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       2.000e+00  6.477e-01   3.088  0.00216 **
## enginebevy       -7.889e-01  6.533e-01  -1.208  0.22794
## enginebevy_extra  3.333e-01  1.058e+00   0.315  0.75283
## enginechuot      -2.000e+00  1.587e+00  -1.261  0.20822
## enginecomfy       1.500e+00  1.212e+00   1.238  0.21652
## enginecustom      4.290e-16  6.729e-01   0.000  1.00000
## enginegamercade   1.000e+00  1.587e+00   0.630  0.52888
## enginegodot      -1.000e+00  8.079e-01  -1.238  0.21652
## engineggez       -1.000e+00  8.770e-01  -1.140  0.25490
## enginehotham     -1.000e+00  1.212e+00  -0.825  0.40975
## engineother       3.000e+00  1.587e+00   1.891  0.05939 .
## enginequad       -6.667e-01  7.479e-01  -0.891  0.37330
## enginenannou     -2.570e-16  1.587e+00   0.000  1.00000
## enginenone       -4.286e-01  8.481e-01  -0.505  0.61360
## enginepiston     -2.000e+00  1.587e+00  -1.261  0.20822
## engineraylib      7.022e-16  1.212e+00   0.000  1.00000
## enginespeedy2d    3.000e+00  1.587e+00   1.891  0.05939 .
## enginetetra      -2.000e+00  1.587e+00  -1.261  0.20822
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.448 on 392 degrees of freedom
## Multiple R-squared:  0.09322,    Adjusted R-squared:  0.0539
## F-statistic: 2.371 on 17 and 392 DF,  p-value: 0.001739
```

```
lm(bad_docs ~ engine, data = dat) |> summary()
```

```
##
## Call:
## lm(formula = bad_docs ~ engine, data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.0000 -1.1626 -0.1626  0.8374  3.6825
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       1.60000    0.70804   2.260   0.0244 *
## enginebevy        0.56263    0.71414   0.788   0.4313
## enginebevy_extra  1.40000    1.15623   1.211   0.2267
## enginechuot      -1.60000    1.73434  -0.923   0.3568
## enginecomfy       3.40000    1.32462   2.567   0.0106 *
## enginecustom     -0.28254    0.73560  -0.384   0.7011
## enginegamercade  -1.60000    1.73434  -0.923   0.3568
## enginegodot      -0.04444    0.88308  -0.050   0.9599
## engineggez        0.40000    0.95869   0.417   0.6767
## enginehotham     -0.10000    1.32462  -0.075   0.9399
```

```
## engineother       2.40000    1.73434   1.384   0.1672
## enginequad        0.86667    0.81758   1.060   0.2898
## enginenannou      3.40000    1.73434   1.960   0.0507 .
## enginenone       -0.88571    0.92704  -0.955   0.3400
## enginepiston     -1.60000    1.73434  -0.923   0.3568
## engineraylib      2.90000    1.32462   2.189   0.0292 *
## enginespeedy2d   -1.60000    1.73434  -0.923   0.3568
## enginetetra      -0.60000    1.73434  -0.346   0.7296
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.583 on 392 degrees of freedom
## Multiple R-squared:  0.1075, Adjusted R-squared:  0.06879
## F-statistic: 2.777 on 17 and 392 DF,  p-value: 0.0002107
```

```
lm(bad_tooling ~ engine, data = dat) |> summary()
```

```
##
## Call:
## lm(formula = bad_tooling ~ engine, data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.0000 -1.4429 -0.4429  1.5571  3.1333
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.000e+00  8.024e-01   3.739 0.000213 ***
## enginebevy       -5.571e-01  8.093e-01  -0.688 0.491652
## enginebevy_extra  1.438e-15  1.310e+00   0.000 1.000000
## enginechuot       1.000e+00  1.966e+00   0.509 0.611202
## enginecomfy      -1.000e+00  1.501e+00  -0.666 0.505719
## enginecustom     -1.111e+00  8.337e-01  -1.333 0.183369
## enginegamercade  -3.000e+00  1.966e+00  -1.526 0.127742
## enginegodot      -1.667e+00  1.001e+00  -1.665 0.096646 .
## engineggez       -1.167e+00  1.086e+00  -1.074 0.283577
## enginehotham      1.272e-15  1.501e+00   0.000 1.000000
## engineother       2.000e+00  1.966e+00   1.018 0.309528
## enginequad       -1.133e+00  9.266e-01  -1.223 0.222005
## enginenannou     -1.000e+00  1.966e+00  -0.509 0.611202
## enginenone       -7.143e-01  1.051e+00  -0.680 0.496990
## enginepiston     -3.000e+00  1.966e+00  -1.526 0.127742
## engineraylib      1.500e+00  1.501e+00   0.999 0.318316
## enginespeedy2d   -3.000e+00  1.966e+00  -1.526 0.127742
## enginetetra      -3.000e+00  1.966e+00  -1.526 0.127742
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.794 on 392 degrees of freedom
## Multiple R-squared:  0.05627,    Adjusted R-squared:  0.01535
## F-statistic: 1.375 on 17 and 392 DF,  p-value: 0.1449
```

```
lm(bad_paying_for_bugs ~ engine, data = dat) |> summary()
```

```
##
```

```
## Call:
## lm(formula = bad_paying_for_bugs ~ engine, data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.0000 -0.5294 -0.5294  0.4706  4.4706
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        0.80000    0.47040   1.701   0.0898 .
## enginebevy        -0.27059    0.47445  -0.570   0.5688
## enginebevy_extra   1.20000    0.76815   1.562   0.1190
## enginechuot       -0.80000    1.15223  -0.694   0.4879
## enginecomfy       -0.30000    0.88003  -0.341   0.7334
## enginecustom      -0.19683    0.48871  -0.403   0.6874
## enginegamercade    0.20000    1.15223   0.174   0.8623
## enginegodot       -0.24444    0.58669  -0.417   0.6772
## engineggez        -0.63333    0.63692  -0.994   0.3207
## enginehotham      -0.30000    0.88003  -0.341   0.7334
## engineother        0.20000    1.15223   0.174   0.8623
## enginequad        -0.60000    0.54317  -1.105   0.2700
## enginenannou       1.20000    1.15223   1.041   0.2983
## enginenone         0.05714    0.61589   0.093   0.9261
## enginepiston      -0.80000    1.15223  -0.694   0.4879
## engineraylib       0.70000    0.88003   0.795   0.4268
## enginespeedy2d    -0.80000    1.15223  -0.694   0.4879
## enginetetra       -0.80000    1.15223  -0.694   0.4879
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.052 on 392 degrees of freedom
## Multiple R-squared:  0.03522,    Adjusted R-squared:  -0.00662
## F-statistic: 0.8418 on 17 and 392 DF,  p-value: 0.6442
```

```
lm(bad_console ~ engine, data = dat) |> summary()
```

```
##
## Call:
## lm(formula = bad_console ~ engine, data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.0000 -1.0208 -0.8889  0.9792  3.9792
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)         1.8000     0.6893   2.611  0.00937 **
## enginebevy         -0.7792     0.6952  -1.121  0.26305
## enginebevy_extra   -0.1333     1.1256  -0.118  0.90577
## enginechuot        -1.8000     1.6884  -1.066  0.28705
## enginecomfy        -1.3000     1.2896  -1.008  0.31404
## enginecustom       -0.1016     0.7161  -0.142  0.88727
## enginegamercade    -1.8000     1.6884  -1.066  0.28705
## enginegodot        -0.9111     0.8597  -1.060  0.28990
## engineggez         -0.4667     0.9333  -0.500  0.61735
```

```
## enginehotham      -1.3000     1.2896  -1.008  0.31404
## engineother        0.2000     1.6884   0.118  0.90577
## enginequad        -0.9333     0.7959  -1.173  0.24166
## enginenannou      -0.8000     1.6884  -0.474  0.63590
## enginenone        -1.3714     0.9025  -1.520  0.12943
## enginepiston      -1.8000     1.6884  -1.066  0.28705
## engineraylib       0.2000     1.2896   0.155  0.87683
## enginespeedy2d    -1.8000     1.6884  -1.066  0.28705
## enginetetra       -0.8000     1.6884  -0.474  0.63590
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.541 on 392 degrees of freedom
## Multiple R-squared:  0.04227,    Adjusted R-squared:  0.0007317
## F-statistic: 1.018 on 17 and 392 DF,  p-value: 0.4376
```

```
lm(bad_mobile ~ engine, data = dat) |> summary()
```

```
##
## Call:
## lm(formula = bad_mobile ~ engine, data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.0000 -1.2526 -1.0000  0.7474  3.8000
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)        2.8000     0.7418   3.775 0.000185 ***
## enginebevy        -1.5474     0.7482  -2.068 0.039279 *
## enginebevy_extra  -2.4667     1.2114  -2.036 0.042396 *
## enginechuot       -0.8000     1.8171  -0.440 0.659983
## enginecomfy       -1.3000     1.3878  -0.937 0.349472
## enginecustom      -1.5143     0.7707  -1.965 0.050137 .
## enginegamercade   -2.8000     1.8171  -1.541 0.124135
## enginegodot       -1.2444     0.9252  -1.345 0.179385
## engineggez        -0.9667     1.0044  -0.962 0.336434
## enginehotham      -2.8000     1.3878  -2.018 0.044316 *
## engineother       -2.8000     1.8171  -1.541 0.124135
## enginequad        -1.6000     0.8566  -1.868 0.062519 .
## enginenannou      -0.8000     1.8171  -0.440 0.659983
## enginenone        -2.3714     0.9713  -2.442 0.015063 *
## enginepiston      -2.8000     1.8171  -1.541 0.124135
## engineraylib       0.2000     1.3878   0.144 0.885485
## enginespeedy2d     2.2000     1.8171   1.211 0.226721
## enginetetra       -0.8000     1.8171  -0.440 0.659983
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.659 on 392 degrees of freedom
## Multiple R-squared:  0.04578,    Adjusted R-squared:  0.004401
## F-statistic: 1.106 on 17 and 392 DF,  p-value: 0.3449
```

```r
lm(bad_web ~ engine, data = dat) |> summary()
```

```
##
## Call:
## lm(formula = bad_web ~ engine, data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.8333 -1.1003 -0.8889  0.8997  4.0635
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)         1.6000     0.6449   2.481   0.0135 *
## enginebevy         -0.4997     0.6504  -0.768   0.4428
## enginebevy_extra   -1.2667     1.0531  -1.203   0.2298
## enginechuot         2.4000     1.5796   1.519   0.1295
## enginecomfy         2.9000     1.2064   2.404   0.0167 *
## enginecustom       -0.6635     0.6700  -0.990   0.3226
## enginegamercade    -1.6000     1.5796  -1.013   0.3117
## enginegodot        -0.7111     0.8043  -0.884   0.3772
## engineggez          0.2333     0.8732   0.267   0.7894
## enginehotham       -0.6000     1.2064  -0.497   0.6192
## engineother         1.4000     1.5796   0.886   0.3760
## enginequad         -0.6667     0.7446  -0.895   0.3712
## enginenannou        1.4000     1.5796   0.886   0.3760
## enginenone         -1.0286     0.8443  -1.218   0.2239
## enginepiston       -1.6000     1.5796  -1.013   0.3117
## engineraylib        0.9000     1.2064   0.746   0.4561
## enginespeedy2d     -1.6000     1.5796  -1.013   0.3117
## enginetetra         1.4000     1.5796   0.886   0.3760
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.442 on 392 degrees of freedom
## Multiple R-squared:  0.06923,    Adjusted R-squared:  0.02887
## F-statistic: 1.715 on 17 and 392 DF,  p-value: 0.03783
```

```r
lm(bad_engine_bugs ~ engine, data = dat) |> summary()
```

```
##
## Call:
## lm(formula = bad_engine_bugs ~ engine, data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.5000 -0.9412  0.0000  0.3175  4.3175
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)         1.6000     0.5235   3.056  0.00239 **
## enginebevy         -0.6588     0.5280  -1.248  0.21285
## enginebevy_extra   -0.2667     0.8548  -0.312  0.75524
## enginechuot        -1.6000     1.2823  -1.248  0.21285
## enginecomfy         0.9000     0.9793   0.919  0.35867
```

```
## enginecustom      -0.9175     0.5439  -1.687  0.09241 .
## enginegamercade   -1.6000     1.2823  -1.248  0.21285
## enginegodot       -1.1556     0.6529  -1.770  0.07752 .
## engineggez        -0.1000     0.7088  -0.141  0.88788
## enginehotham      -0.1000     0.9793  -0.102  0.91872
## engineother        2.4000     1.2823   1.872  0.06199 .
## enginequad        -0.3333     0.6045  -0.551  0.58164
## enginenannou      -0.6000     1.2823  -0.468  0.64010
## enginenone        -1.3143     0.6854  -1.918  0.05589 .
## enginepiston      -1.6000     1.2823  -1.248  0.21285
## engineraylib      -1.1000     0.9793  -1.123  0.26204
## enginespeedy2d    -1.6000     1.2823  -1.248  0.21285
## enginetetra       -1.6000     1.2823  -1.248  0.21285
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.171 on 392 degrees of freedom
## Multiple R-squared:  0.06103,    Adjusted R-squared:  0.02031
## F-statistic: 1.499 on 17 and 392 DF,  p-value: 0.09133
```

```
lm(bad_engine_features ~ engine, data = dat) |> summary()
```

```
##
## Call:
## lm(formula = bad_engine_features ~ engine, data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7647 -1.4444  0.2353  1.2353  3.5556
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       2.000e+00  7.282e-01   2.746   0.0063 **
## enginebevy        7.647e-01  7.345e-01   1.041   0.2984
## enginebevy_extra  3.333e-01  1.189e+00   0.280   0.7794
## enginechuot      -2.000e+00  1.784e+00  -1.121   0.2629
## enginecomfy      -5.000e-01  1.362e+00  -0.367   0.7138
## enginecustom     -5.556e-01  7.566e-01  -0.734   0.4632
## enginegamercade  -2.000e+00  1.784e+00  -1.121   0.2629
## enginegodot       1.111e-01  9.082e-01   0.122   0.9027
## engineggez        1.000e+00  9.860e-01   1.014   0.3111
## enginehotham     -5.000e-01  1.362e+00  -0.367   0.7138
## engineother       3.000e+00  1.784e+00   1.682   0.0934 .
## enginequad        3.333e-01  8.409e-01   0.396   0.6920
## enginenannou      1.000e+00  1.784e+00   0.561   0.5754
## enginenone       -2.857e-01  9.535e-01  -0.300   0.7646
## enginepiston     -2.000e+00  1.784e+00  -1.121   0.2629
## engineraylib      5.000e-01  1.362e+00   0.367   0.7138
## enginespeedy2d    2.000e+00  1.784e+00   1.121   0.2629
## enginetetra      -2.189e-15  1.784e+00   0.000   1.0000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.628 on 392 degrees of freedom
## Multiple R-squared:  0.1119, Adjusted R-squared:  0.07336
```

```
## F-statistic: 2.905 on 17 and 392 DF,  p-value: 0.0001065
```

```
lm(bad_hiring ~ engine, data = dat) |> summary()
```

```
##
## Call:
## lm(formula = bad_hiring ~ engine, data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.4000 -0.5709 -0.5709  0.4291  4.4291
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)         1.4000     0.5603   2.499   0.0129 *
## enginebevy         -0.8291     0.5651  -1.467   0.1432
## enginebevy_extra   -0.4000     0.9150  -0.437   0.6622
## enginechuot        -1.4000     1.3724  -1.020   0.3083
## enginecomfy        -0.4000     1.0482  -0.382   0.7030
## enginecustom       -0.4635     0.5821  -0.796   0.4264
## enginegamercade     2.6000     1.3724   1.894   0.0589 .
## enginegodot        -0.5111     0.6988  -0.731   0.4650
## engineggez         -1.2333     0.7586  -1.626   0.1048
## enginehotham       -1.4000     1.0482  -1.336   0.1825
## engineother        -1.4000     1.3724  -1.020   0.3083
## enginequad         -0.7333     0.6470  -1.133   0.2577
## enginenannou       -1.4000     1.3724  -1.020   0.3083
## enginenone         -0.4000     0.7336  -0.545   0.5859
## enginepiston       -1.4000     1.3724  -1.020   0.3083
## engineraylib       -0.9000     1.0482  -0.859   0.3911
## enginespeedy2d     -1.4000     1.3724  -1.020   0.3083
## enginetetra        -1.4000     1.3724  -1.020   0.3083
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.253 on 392 degrees of freedom
## Multiple R-squared:  0.04327,    Adjusted R-squared:  0.001777
## F-statistic: 1.043 on 17 and 392 DF,  p-value: 0.4101
```

```
lm(bad_performance ~ engine, data = dat) |> summary()
```

```
##
## Call:
## lm(formula = bad_performance ~ engine, data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.0000 -0.6678 -0.3810  0.3322  4.3322
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)         1.0000     0.4661   2.145 0.032534 *
## enginebevy         -0.3322     0.4701  -0.707 0.480250
## enginebevy_extra   -0.6667     0.7612  -0.876 0.381643
## enginechuot        -1.0000     1.1417  -0.876 0.381643
```

```
## enginecomfy        1.0000     0.8720   1.147 0.252177
## enginecustom      -0.6190     0.4843  -1.278 0.201882
## enginegamercade   -1.0000     1.1417  -0.876 0.381643
## enginegodot       -1.0000     0.5813  -1.720 0.086194 .
## engineggez        -0.8333     0.6311  -1.320 0.187469
## enginehotham       0.5000     0.8720   0.573 0.566713
## engineother        4.0000     1.1417   3.503 0.000512 ***
## enginequad        -0.9333     0.5382  -1.734 0.083684 .
## enginenannou      -1.0000     1.1417  -0.876 0.381643
## enginenone        -0.7143     0.6103  -1.170 0.242544
## enginepiston      -1.0000     1.1417  -0.876 0.381643
## engineraylib      -0.5000     0.8720  -0.573 0.566713
## enginespeedy2d    -1.0000     1.1417  -0.876 0.381643
## enginetetra       -1.0000     1.1417  -0.876 0.381643
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.042 on 392 degrees of freedom
## Multiple R-squared:  0.08916,    Adjusted R-squared:  0.04966
## F-statistic: 2.257 on 17 and 392 DF,  p-value: 0.003064
```

There are imo no significant correlations, considering the huge number of tests and the small sample sizes for some of these categories.