Exp 1: Difficulty

Setup

Load Data

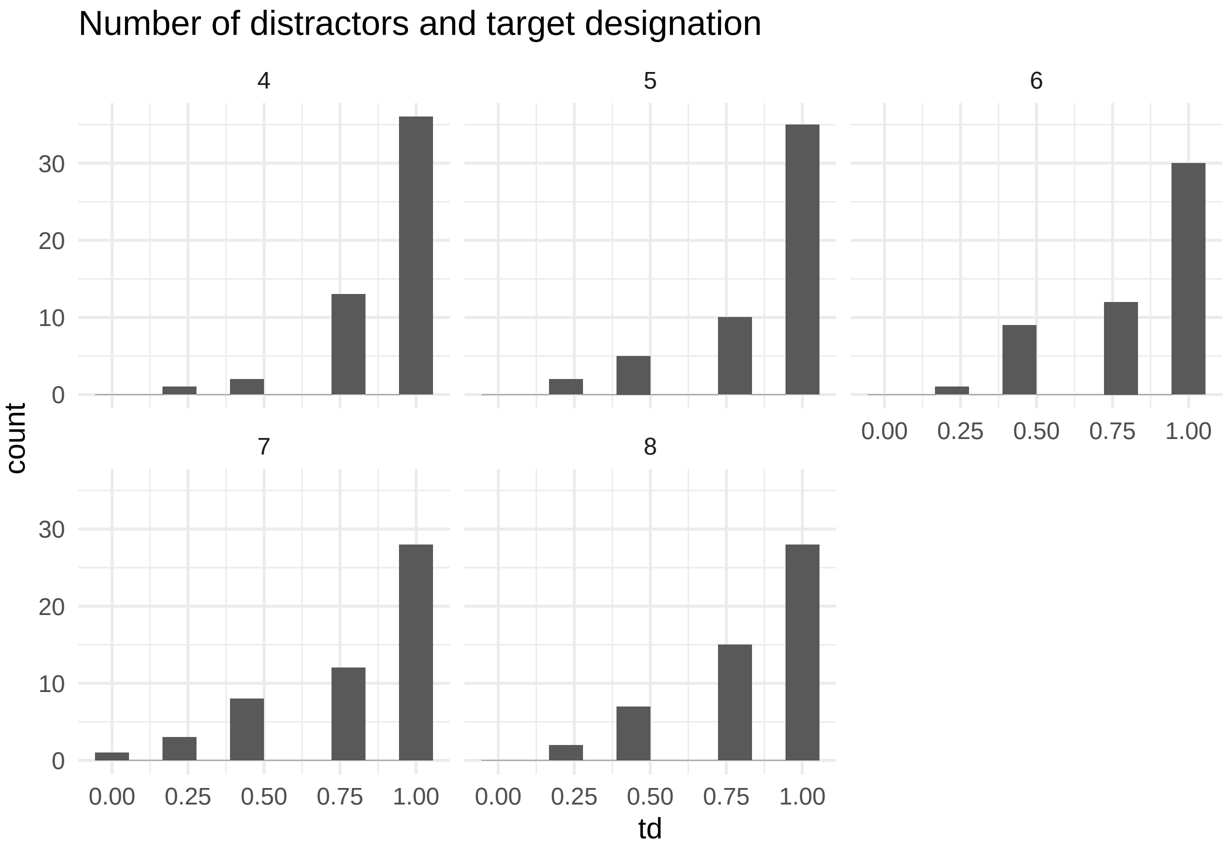
```
subject_data <- read_csv("data/exp1_difficulty_parsed_trials.csv") %>%
  select(-c(WID)) %>%
  replace_na(list(response_frame = Inf)) %>%
  group_by(ID) %>%
  mutate(difficulty = scale(difficulty))

model_data <- read_csv("data/exp1_difficulty_target_designation.csv")
```

```
td_by_subj_tracker <- subject_data %>%
  pivot_longer(cols = starts_with("td"),
               names_to = "tracker",
               values_to = "td") %>%
  separate(tracker, c("NA", "tracker")) %>%
  mutate(tracker = as.numeric(tracker)) %>%
  group_by(ID, scene, tracker) %>%
  summarise(td = first(td),
            vel = first(vel),
            n_dist = first(n_dist),
            difficulty = first(difficulty))

td_by_subj <- td_by_subj_tracker %>%
  group_by(ID) %>%
  summarise(td_acc_mu = mean(td),
            n = n(),
            td_acc_se = sd(td) / sqrt(n),
            passed = td_acc_mu > 0.5 + 3 * td_acc_se) # chance performance lower than 0.5?
td_by_subj
```

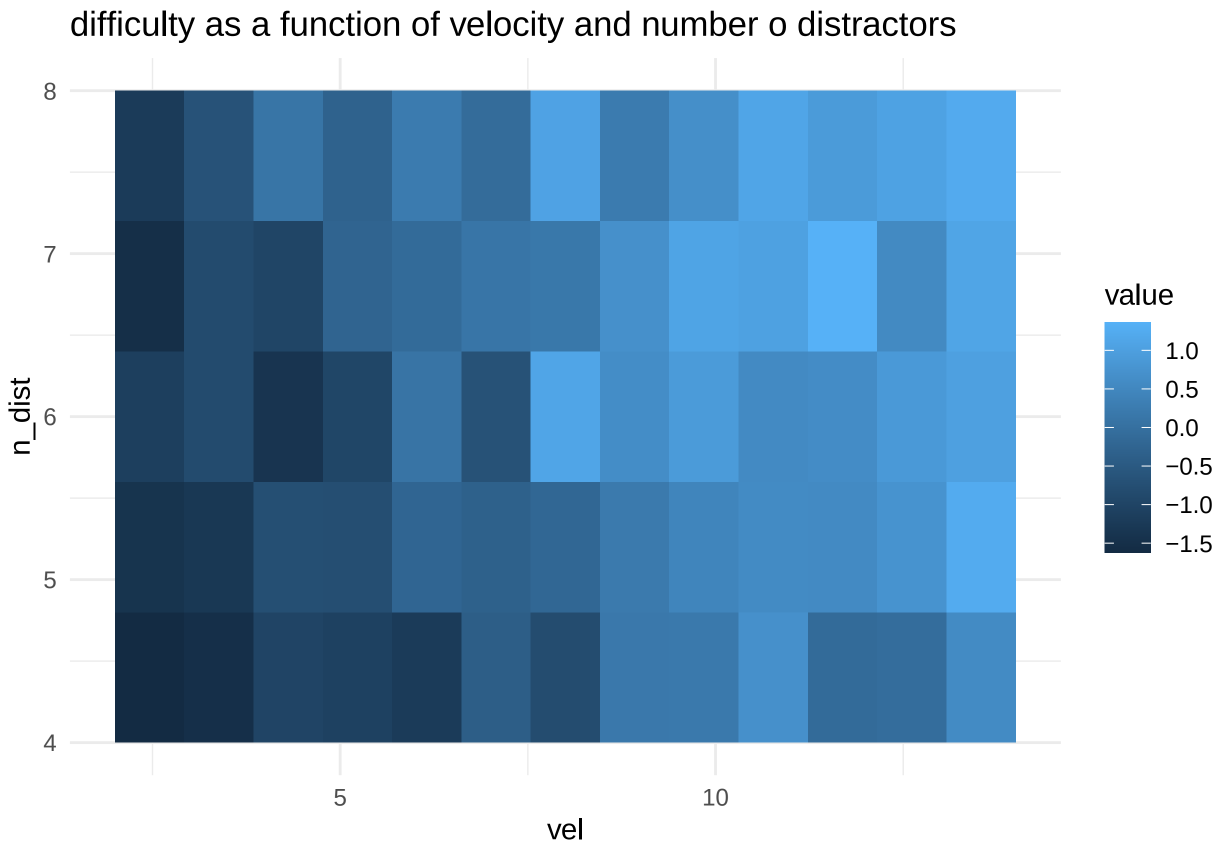
```
## # A tibble: 5 x 5
##      ID td_acc_mu     n td_acc_se passed
##   <dbl>     <dbl> <int>     <dbl> <lgl>
## 1     0     0.542   260    0.0310 FALSE
## 2     1     0.819   260    0.0239 TRUE
## 3     2     0.9     260    0.0186 TRUE
## 4     3     0.812   260    0.0243 TRUE
## 5     4     0.873   260    0.0207 TRUE
```

```
good_td_by_subj_tracker <- td_by_subj %>%
  filter(passed) %>%
  left_join(td_by_subj_tracker)


good_td_by_scene <- good_td_by_subj_tracker %>%
  group_by(scene) %>%
  summarise(td = mean(td),
            vel = first(vel),
            n_dist = first(n_dist),
            difficulty = mean(difficulty))

good_td_by_subj_scene <- good_td_by_subj_tracker %>%
  group_by(ID, scene) %>%
  summarise(td = mean(td),
            difficulty = first(difficulty),
            n_dist = first(n_dist))
```

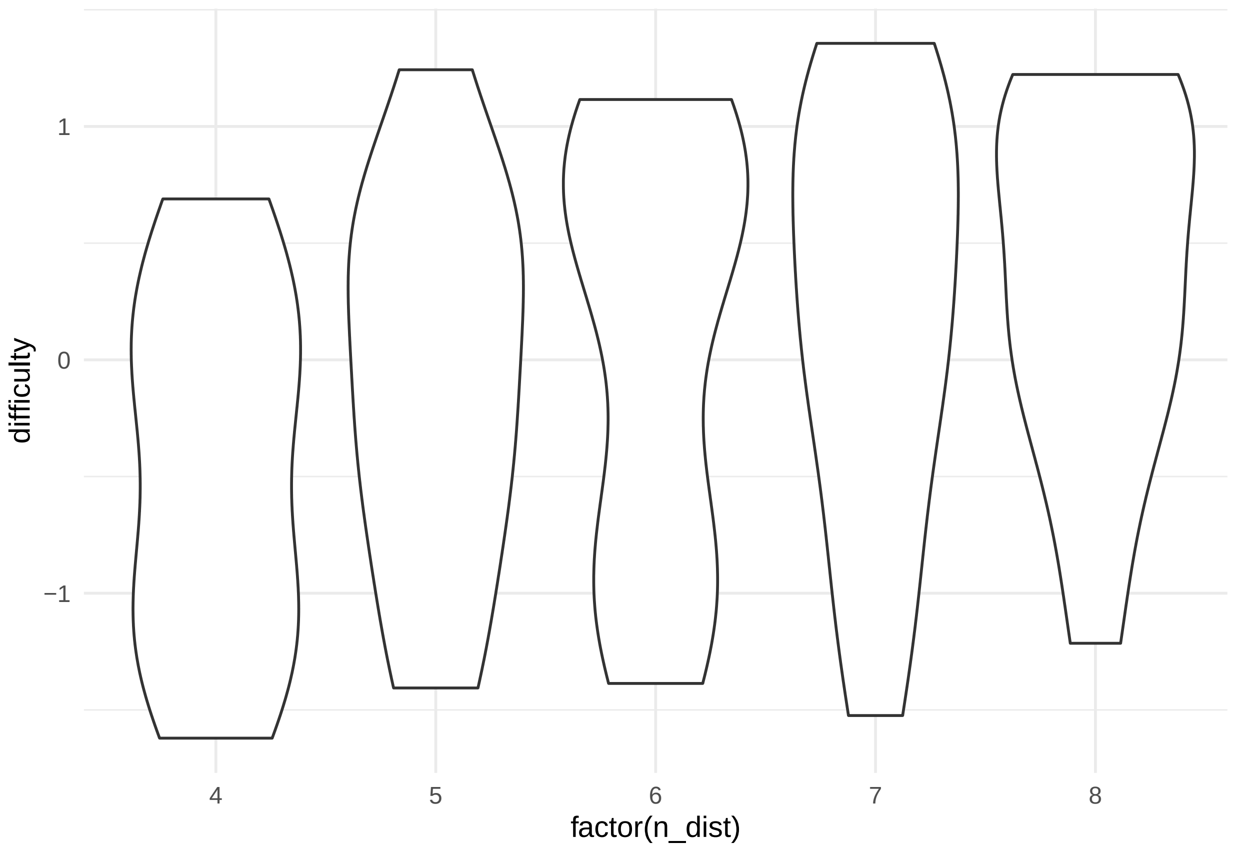
```
good_td_by_subj_scene %>%
  ggplot(aes(x=td)) +
  geom_histogram(bins=10) +
  facet_wrap(vars(n_dist)) +
  ggtitle("Number of distractors and target designation")
```



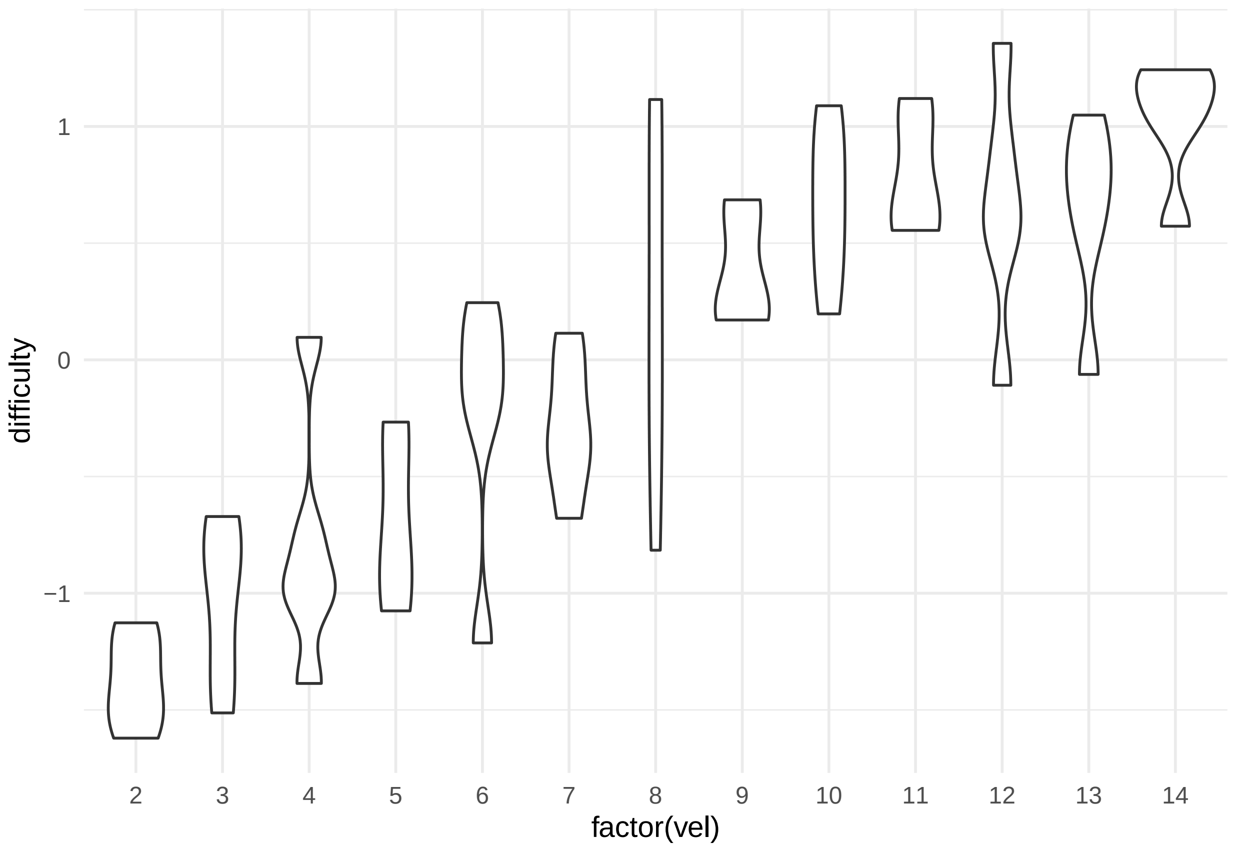
```r
x_breaks <- seq(2.0,14.0,length.out=14)
y_breaks <- seq(4,8,length.out=6)

breaks<-list(x=x_breaks, y=y_breaks)

good_td_by_scene %>%
  ggplot(aes(x=vel, y=n_dist, z=difficulty)) +
  stat_summary_2d(breaks=breaks) +
  ggtitle("difficulty as a function of velocity and number o distractors")
```



```r
good_td_by_scene %>%
  ggplot(aes(factor(n_dist), y=difficulty)) +
  geom_violin()
```



```
good_td_by_scene %>%
  ggplot(aes(factor(vel), y=difficulty)) +
  geom_violin()
```

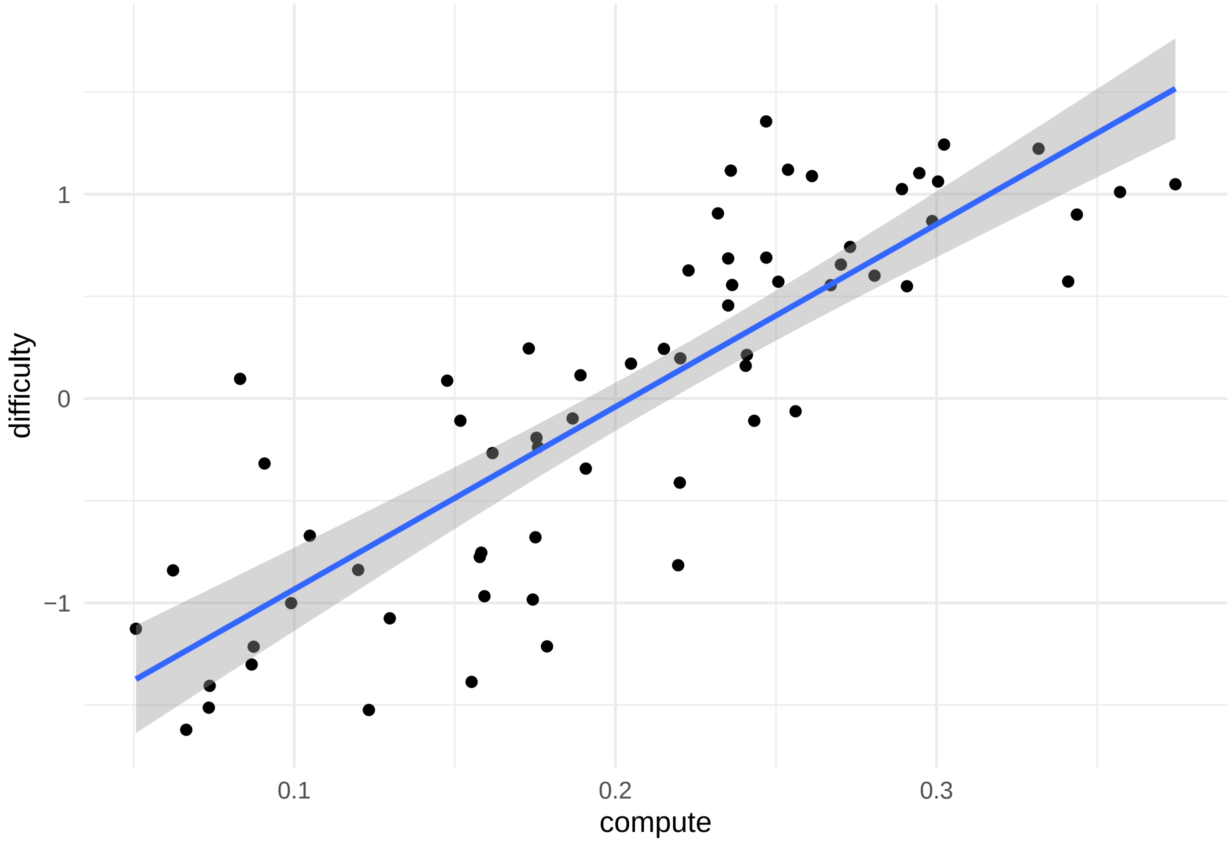

1
0
−1
difficulty
2
3
4
5
6
7
8
9
10
11
12
13
14
factor(vel)

Model compute explaining difficulty

```
model_td_compute_by_scene <- model_data %>%
  group_by(scene) %>%
  summarize(td_model = mean(td_acc),
            compute = mean(attention))

good_full_data <- good_td_by_scene %>%
  left_join(model_td_compute_by_scene)

good_full_data %>%
  with(lm_robust(difficulty ~ compute)) %>%
  summary()
```

```
##
## Call:
## lm_robust(formula = difficulty ~ compute)
##
## Standard error type:  HC2
##
## Coefficients:
##             Estimate Std. Error t value  Pr(>|t|) CI Lower CI Upper DF
## (Intercept)   -1.826     0.1649  -11.08 1.992e-16   -2.156   -1.497 63
## compute        8.929     0.7036   12.69 5.163e-19    7.523   10.336 63
##
## Multiple R-squared:  0.7028 ,    Adjusted R-squared:  0.6981
## F-statistic: 161.1 on 1 and 63 DF,  p-value: < 2.2e-16
```

```
good_full_data %>%
  ggplot(aes(x=compute, y=difficulty)) +
  geom_point() +
  geom_smooth(method="lm_robust")
```

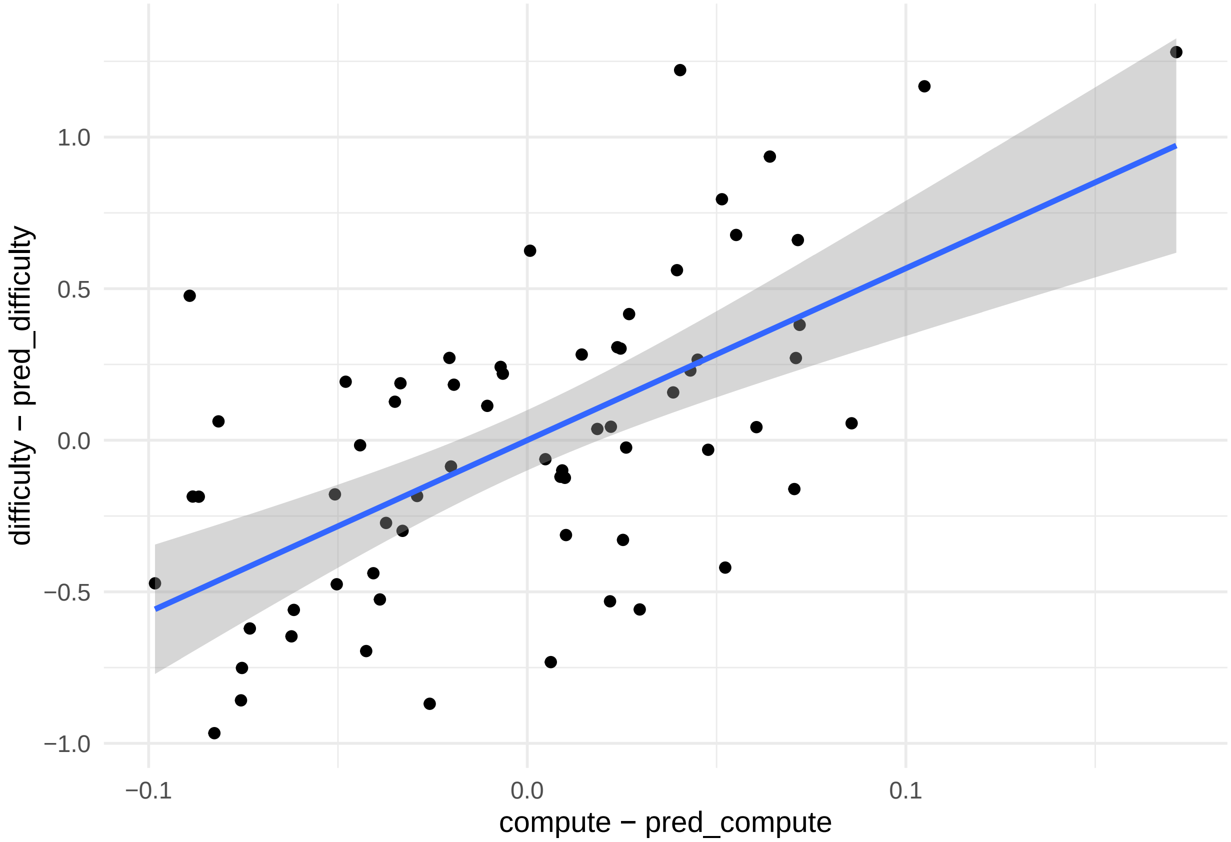


Residualized analysis

```
pred_compute <- good_full_data %>%
  with(lm_robust(compute ~ td)) %>%
  predict()
pred_difficulty <- good_full_data %>%
  with(lm_robust(difficulty ~ td)) %>%
  predict()

good_full_data %>%
  with(lm_robust(difficulty - pred_difficulty ~ compute - pred_compute)) %>%
  summary()
```

```
##
## Call:
## lm_robust(formula = difficulty - pred_difficulty ~ compute -
##     pred_compute)
##
## Standard error type:  HC2
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|) CI Lower CI Upper DF
## (Intercept)  -0.5424     0.1647  -3.292 0.001631  -0.8716  -0.2132 63
## compute       2.6521     0.8363   3.171 0.002345   0.9809   4.3233 63
##
## Multiple R-squared:  0.1784 ,    Adjusted R-squared:  0.1653
## F-statistic: 10.06 on 1 and 63 DF,  p-value: 0.002345
```

```
good_full_data %>%
  ggplot(aes(x=compute - pred_compute, y=difficulty-pred_difficulty)) +
  geom_point() +
  geom_smooth(method="lm_robust")
```


difficulty − pred_difficulty
compute − pred_compute