Course 22160: TA position task

Jonas Dalsberg Jørgensen (s213551)

2023-07-24

Contents


```
# Loading packages
library("tidyverse")
library("broom")
```

Data wrangling

```
# Loading the raw Gravier data
gravier_raw <- read_rds(file = "gravier.rdata")

# 1-4: Creating a Gravier tibble with relocated and recoded "y"
gravier_clean <- gravier_raw %>%
  bind_cols %>%
  as_tibble %>%
  relocate(y) %>%
  rename(outcome = y) %>%
  mutate(outcome = case_when(outcome == "good" ~ 0,
                             outcome == "poor" ~ 1))

gravier_clean
```

```
## # A tibble: 168 x 2,906
##    outcome    g2E09    g7F07    g1A01   g3C09      g3H08    g1A08    g1B01
##      <dbl>    <dbl>    <dbl>    <dbl>   <dbl>      <dbl>    <dbl>    <dbl>
##  1       0 -0.00144 -0.00144 -0.0831  -0.0475  0.0158    -0.0336  -0.136
##  2       0 -0.0604   0.0129  -0.00144  0.0104  0.0316     0.108    0.0158
##  3       0  0.0398   0.0524  -0.0786   0.0635 -0.0395     0.0342   0.00288
##  4       0  0.0101   0.0314  -0.0218   0.0215  0.0868     0.0272  -0.0160
##  5       0  0.0496   0.0201   0.0370   0.0311  0.0207    -0.0174   0.111
##  6       0 -0.0664   0.0468   0.00720 -0.370   0.00288    0.0243   0.0909
##  7       0 -0.00289 -0.0816  -0.0291  -0.0249 -0.0174     0.0172  -0.170
##  8       0 -0.198   -0.0499  -0.0634  -0.0298  0.0300     0.00144 -0.0529
##  9       0  0.00288  0.0201   0.0272   0.0174 -0.0000789 -0.0634   0.0370
## 10       0 -0.0574  -0.0574  -0.0831  -0.0897 -0.101     -0.144   -0.167
## # ... with 158 more rows, and 2,898 more variables: g1int1 <dbl>, g1E11 <dbl>,
## #   g8G02 <dbl>, g1H04 <dbl>, g1C01 <dbl>, g1F11 <dbl>, g3F05 <dbl>,
## #   g3B09 <dbl>, g1int2 <dbl>, g2C01 <dbl>, g1A05 <dbl>, g1E01 <dbl>,
```

```
## #   g1B05 <dbl>, g3C05 <dbl>, g3A07 <dbl>, g1F01 <dbl>, g2D01 <dbl>,
## #   g1int3 <dbl>, g1int4 <dbl>, g1D05 <dbl>, g1E05 <dbl>, g1G05 <dbl>,
## #   g1C05 <dbl>, g1G11 <dbl>, g2D08 <dbl>, g2E06 <dbl>, g3H09 <dbl>,
## #   g2F09 <dbl>, g3G06 <dbl>, g2G08 <dbl>, g3F07 <dbl>, g2G09 <dbl>, ...
```

```
# 5: Reformatting the data to a long format
gravier_data_long <- gravier_clean %>%
  pivot_longer(cols = -outcome,
               names_to = "gene",
               values_to = "log2_expr_level")

gravier_data_long
```

```
## # A tibble: 488,040 x 3
##    outcome gene   log2_expr_level
##      <dbl> <chr>            <dbl>
##  1       0 g2E09          -0.00144
##  2       0 g7F07          -0.00144
##  3       0 g1A01          -0.0831
##  4       0 g3C09          -0.0475
##  5       0 g3H08           0.0158
##  6       0 g1A08          -0.0336
##  7       0 g1B01          -0.136
##  8       0 g1int1          0.0180
##  9       0 g1E11           0.0257
## 10       0 g8G02           0.00720
## # ... with 488,030 more rows
```

```
# Creating a nested tibble of outcome and gene expression level for each gene for modelling purposes
gravier_data_long_nested <- gravier_data_long %>%
  group_by(gene) %>%
  nest() %>%
  ungroup()

gravier_data_long_nested
```

```
## # A tibble: 2,905 x 2
##    gene   data
##    <chr>  <list>
##  1 g2E09  <tibble [168 x 2]>
##  2 g7F07  <tibble [168 x 2]>
##  3 g1A01  <tibble [168 x 2]>
##  4 g3C09  <tibble [168 x 2]>
##  5 g3H08  <tibble [168 x 2]>
##  6 g1A08  <tibble [168 x 2]>
##  7 g1B01  <tibble [168 x 2]>
##  8 g1int1 <tibble [168 x 2]>
##  9 g1E11  <tibble [168 x 2]>
## 10 g8G02  <tibble [168 x 2]>
## # ... with 2,895 more rows
```

```r
# 6: Randomly selecting 100 genes
set.seed(42)
gravier_data_long_nested_100 <- gravier_data_long_nested %>%
  sample_n(100)

gravier_data_long_nested_100
```

```
## # A tibble: 100 x 2
##    gene      data
##    <chr>     <list>
##  1 g1int1611 <tibble [168 x 2]>
##  2 g8H12     <tibble [168 x 2]>
##  3 g1int707  <tibble [168 x 2]>
##  4 g7C09     <tibble [168 x 2]>
##  5 g5A03     <tibble [168 x 2]>
##  6 g1CNS585  <tibble [168 x 2]>
##  7 g1int1296 <tibble [168 x 2]>
##  8 g1int690  <tibble [168 x 2]>
##  9 g1int796  <tibble [168 x 2]>
## 10 g1int1277 <tibble [168 x 2]>
## # ... with 90 more rows
```

```r
# 7: Fitting a logistic regression model to each gene
gravier_data_long_nested_100 <- gravier_data_long_nested_100 %>%
  mutate(mdl = map(data,
                   ~glm(outcome ~ log2_expr_level,
                        data = .,
                        family = binomial(link = "logit")),
                   conf.int = TRUE))

gravier_data_long_nested_100
```

```
## # A tibble: 100 x 3
##    gene      data               mdl
##    <chr>     <list>             <list>
##  1 g1int1611 <tibble [168 x 2]> <glm>
##  2 g8H12     <tibble [168 x 2]> <glm>
##  3 g1int707  <tibble [168 x 2]> <glm>
##  4 g7C09     <tibble [168 x 2]> <glm>
##  5 g5A03     <tibble [168 x 2]> <glm>
##  6 g1CNS585  <tibble [168 x 2]> <glm>
##  7 g1int1296 <tibble [168 x 2]> <glm>
##  8 g1int690  <tibble [168 x 2]> <glm>
##  9 g1int796  <tibble [168 x 2]> <glm>
## 10 g1int1277 <tibble [168 x 2]> <glm>
## # ... with 90 more rows
```

```r
# 8: Add beta-estimates and confidence intervals

# Extracting information from the models
gravier_data_long_nested_100 <- gravier_data_long_nested_100 %>%
```

```
  mutate(mdl_tidy = map(mdl,
                        ~tidy(.x,
                              # include confidence intervals (default value 0.95):
                              conf.int = TRUE))) %>%
  unnest(mdl_tidy)

# Removing intercept rows and unnecessary columns
gravier_data_long_nested_100 <- gravier_data_long_nested_100 %>%
  filter(term != "(Intercept)") %>%
  select(-std.error,
         -statistic)

gravier_data_long_nested_100
```

```
## # A tibble: 100 x 8
##    gene      data               mdl    term      estim~1 p.value conf.~2 conf.~3
##    <chr>     <list>             <list> <chr>       <dbl>   <dbl>   <dbl>   <dbl>
##  1 g1int1611 <tibble [168 x 2]> <glm>  log2_exp~  -0.711  0.494  -2.82      1.28
##  2 g8H12     <tibble [168 x 2]> <glm>  log2_exp~   0.820  0.504  -1.56      3.32
##  3 g1int707  <tibble [168 x 2]> <glm>  log2_exp~   3.03   0.0613 -0.0451    6.38
##  4 g7C09     <tibble [168 x 2]> <glm>  log2_exp~   3.48   0.0578 -0.0136    7.25
##  5 g5A03     <tibble [168 x 2]> <glm>  log2_exp~   0.325  0.792  -2.13      2.77
##  6 g1CNS585  <tibble [168 x 2]> <glm>  log2_exp~  -0.467  0.666  -2.63      1.65
##  7 g1int1296 <tibble [168 x 2]> <glm>  log2_exp~   1.80   0.0626 -0.0387    3.79
##  8 g1int690  <tibble [168 x 2]> <glm>  log2_exp~   3.23   0.0367  0.306     6.41
##  9 g1int796  <tibble [168 x 2]> <glm>  log2_exp~   4.26   0.0300  0.582     8.34
## 10 g1int1277 <tibble [168 x 2]> <glm>  log2_exp~  -0.307  0.798  -2.66      2.07
## # ... with 90 more rows, and abbreviated variable names 1: estimate,
## #   2: conf.low, 3: conf.high
```

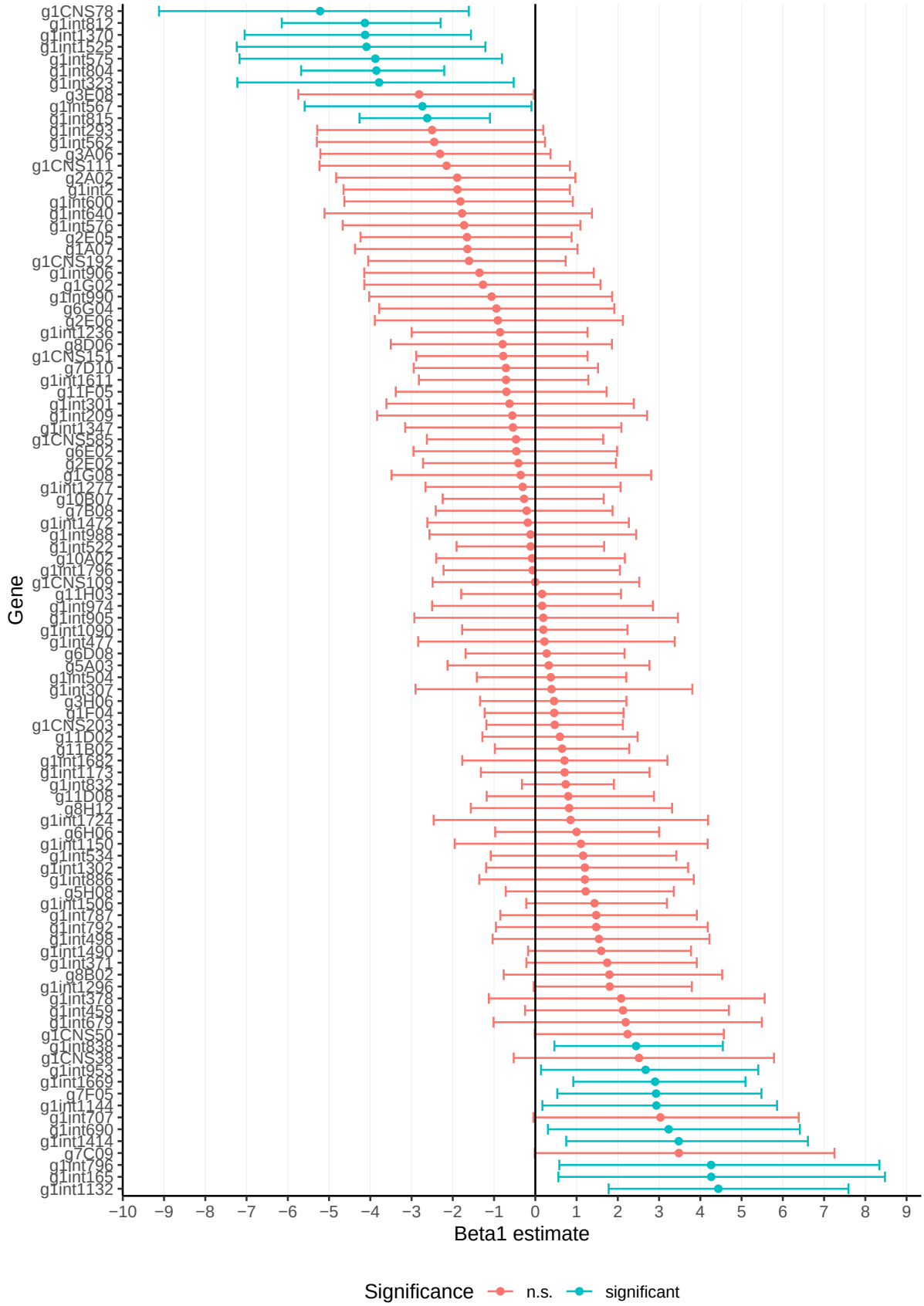
```
# 9: Add indicator for p-value <= 0.05
gravier_data_long_nested_100 <- gravier_data_long_nested_100 %>%
  mutate(is_significant = case_when(p.value <= 0.05 ~ "significant",
                                    p.value > 0.05 ~ "n.s."))

gravier_data_long_nested_100
```

```
## # A tibble: 100 x 9
##    gene      data     mdl    term        estim~1 p.value conf.~2 conf.~3 is_si~4
##    <chr>     <list>   <list> <chr>         <dbl>   <dbl>   <dbl>   <dbl> <chr>
##  1 g1int1611 <tibble> <glm>  log2_expr_~  -0.711  0.494  -2.82      1.28 n.s.
##  2 g8H12     <tibble> <glm>  log2_expr_~   0.820  0.504  -1.56      3.32 n.s.
##  3 g1int707  <tibble> <glm>  log2_expr_~   3.03   0.0613 -0.0451    6.38 n.s.
##  4 g7C09     <tibble> <glm>  log2_expr_~   3.48   0.0578 -0.0136    7.25 n.s.
##  5 g5A03     <tibble> <glm>  log2_expr_~   0.325  0.792  -2.13      2.77 n.s.
##  6 g1CNS585  <tibble> <glm>  log2_expr_~  -0.467  0.666  -2.63      1.65 n.s.
##  7 g1int1296 <tibble> <glm>  log2_expr_~   1.80   0.0626 -0.0387    3.79 n.s.
##  8 g1int690  <tibble> <glm>  log2_expr_~   3.23   0.0367  0.306     6.41 signif~
##  9 g1int796  <tibble> <glm>  log2_expr_~   4.26   0.0300  0.582     8.34 signif~
## 10 g1int1277 <tibble> <glm>  log2_expr_~  -0.307  0.798  -2.66      2.07 n.s.
## # ... with 90 more rows, and abbreviated variable names 1: estimate,
## #   2: conf.low, 3: conf.high, 4: is_significant
```

Plotting

```
# 10-11: Create forest-plot of slopes with 95% CI
gravier_data_long_nested_100 %>%
  arrange(desc(estimate)) %>%
  mutate(gene=factor(gene, levels=gene)) %>%
  ggplot(mapping = aes(x = estimate,
                       y = gene,
                       col = is_significant)) +
  geom_point() +
  geom_errorbarh(mapping = aes(xmin = conf.low,
                               xmax = conf.high)) +
  geom_vline(xintercept = 0) +
  theme_classic() +
  scale_x_continuous(breaks = seq(from = -10,
                                  to = 10,
                                  by = 1)) +
  theme(legend.position = "bottom",
        panel.grid.major.x = element_line(linewidth = .05,
                                          color = "#EEEEEE")) +
  labs(col = "Significance") +
  xlab("Beta1 estimate") +
  ylab("Gene")
```


Gene
g1CNS78
g1int812
g1int1370
g1int1525
g1int575
g1int804
g1int323
g3E08
g1int567
g1int815
g1int293
g1int562
g3A06
g1CNS111
g2A02
g1int2
g1int600
g1int640
g1int576
g2E05
g1A07
g1CNS192
g1int906
g1G02
g1int990
g6G04
g2E06
g1int1236
g8D06
g1CNS151
g7D10
g1int1611
g11F05
g1int301
g1int209
g1int1347
g1CNS585
g6E02
g2E02
g1G08
g1int1277
g10B07
g7B08
g1int1472
g1int988
g1int522
g10A02
g1int1796
g1CNS109
g11H03
g1int974
g1int905
g1int1090
g1int477
g6D08
g5A03
g1int504
g1int307
g3H06
g1F04
g1CNS203
g11D02
g11B02
g1int1682
g1int1173
g1int832
g11D08
g8H12
g1int1724
g6H06
g1int1150
g1int534
g1int1302
g1int886
g5H08
g1int1506
g1int787
g1int792
g1int498
g1int1490
g1int371
g8B02
g1int1296
g1int378
g1int459
g1int679
g1CNS50
g1int838
g1CNS38
g1int953
g1int1669
g7F05
g1int1144
g1int707
g1int690
g1int1414
g7C09
g1int796
g1int165
g1int1132
−10 −9 −8 −7 −6 −5 −4 −3 −2 −1 0 1 2 3 4 5 6 7 8 9
Beta1 estimate
Significance
n.s.
significant