8-analyze-second-run.R

marskar

Thu Apr 19 19:23:01 2018

```
library(readr)
library(here)
```

```
## here() starts at /Users/marskar/gdrive/nhanes
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(purrr)

#define function needed to calculate median model stats
get_median <- function(x, model_type, model_stat){
    model_type <- deparse(substitute(model_type))
    model_stat <- enquo(model_stat)
    x %>%
        select(type, !!model_stat) %>%
        group_by(type) %>%
        summarise(model_median =
                  median(!!model_stat)) %>%
        filter(type == model_type) %>%
        select(model_median) %>%
        as.numeric
}

#read in dataset created by script 4
dat_quad <- read_rds(here("dat/5-model-second-run.rds")) %>%
    rename(con = concordance) %>%
    mutate(quad =
           as.factor(
               case_when(con > median(con) &
                         aic <= median(aic) ~ 1,
                         con > median(con) &
                         aic > median(aic) ~ 2,
                         con <= median(con) &
                         aic <= median(aic) ~ 3,
                         con <= median(con) &
                         aic > median(aic) ~ 4
                        )
```

```
                    )
          )

table(dat_quad$quad)
```

```
##
##   1   2   3   4
##  63  76 337 324
```

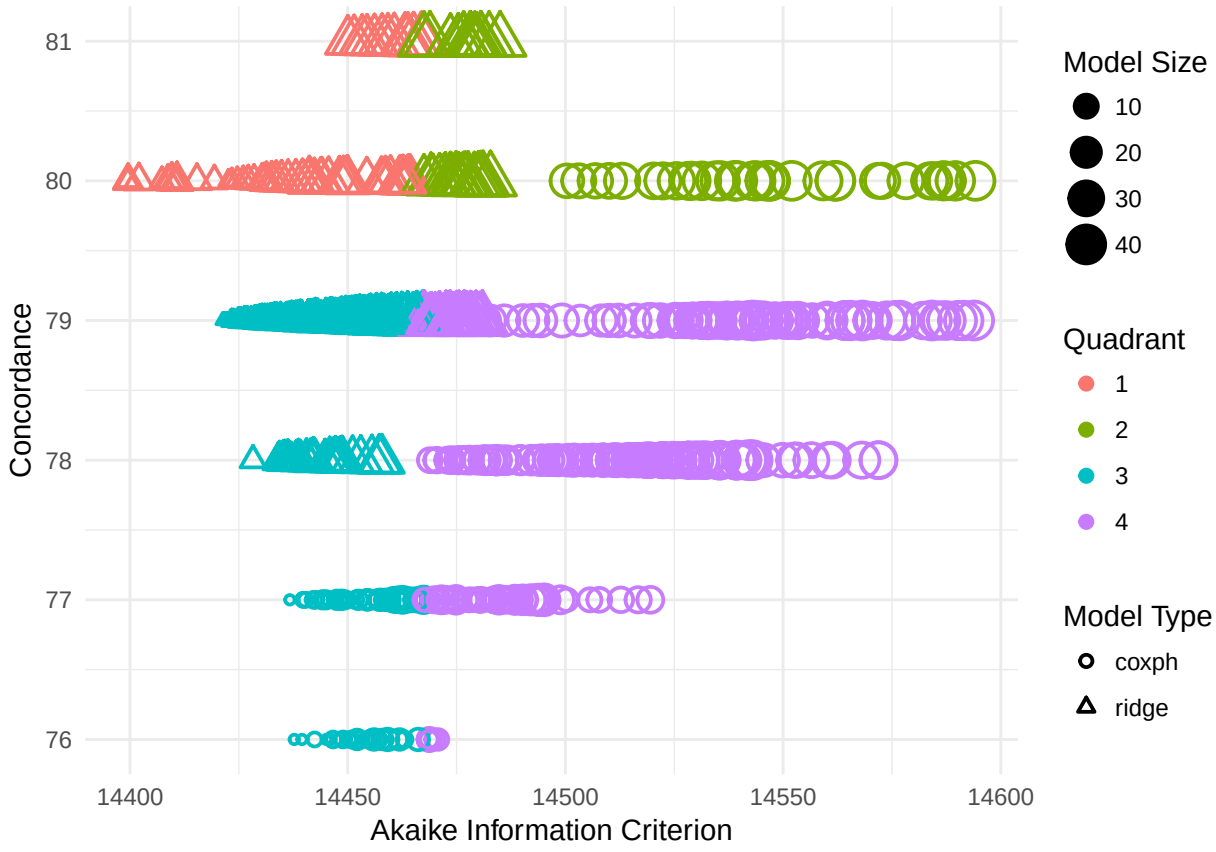
```
table(dat_quad$type)
```

```
##
## coxph ridge
##   400   400
```

```
dat_quad %>% group_by(type, quad) %>% summarise(n=n())
```

```
## # A tibble: 7 x 3
## # Groups:   type [?]
##   type  quad      n
##   <chr> <fct> <int>
## 1 coxph 2        37
## 2 coxph 3        85
## 3 coxph 4       278
## 4 ridge 1        63
## 5 ridge 2        39
## 6 ridge 3       252
## 7 ridge 4        46
```

```
# Figure 1
dat_quad %>%
    ggplot(aes(x = aic,
               y = con,
               size = size,
               colour = quad)) +
geom_point(aes(shape = factor(type)),
           #size = 3,
           stroke = 1) +
scale_shape(solid = FALSE) +
           theme_minimal() +
           labs(
                x = 'Akaike Information Criterion',
                y = 'Concordance',
                size = "Model Size",
                shape = "Model Type",
                colour = "Quadrant")
```




```
ggsave(here("img/1-quad2.pdf"))
```

```
## Saving 6.5 x 4.5 in image
```

```
ggsave(here("img/1-quad2.png"))
```

```
## Saving 6.5 x 4.5 in image
```

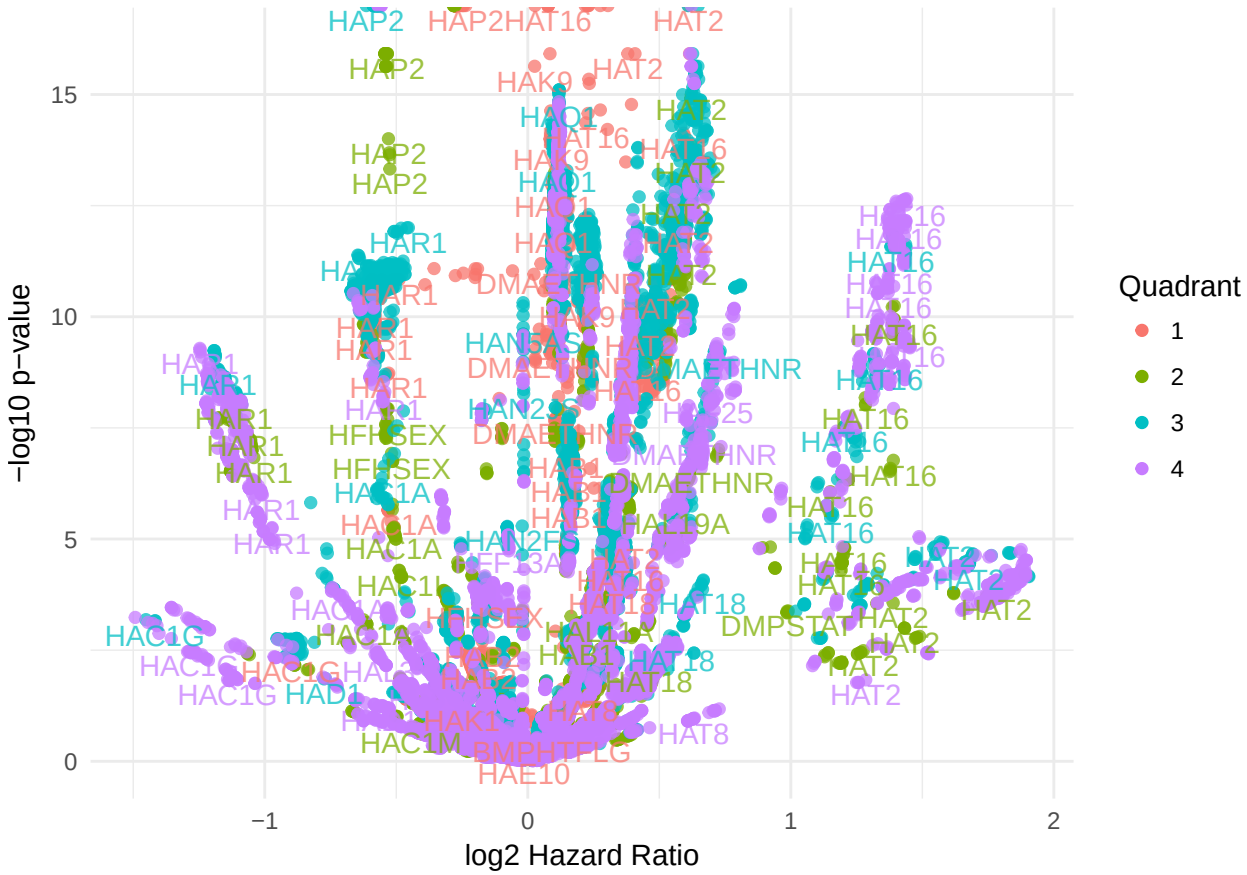
```
#define function to flatten dat_quad
dfs <- function(quadrant) {
dat <- dat_quad %>%
        filter(quad == quadrant) %>%
            select(starts_with('h'),
                   coef_pvalue)

data_frame(name = names(flatten(dat[[1]])),
           HR = flatten_dbl(dat[[1]]),
           HR_CI_lower = flatten_dbl(dat[[2]]),
           HR_CI_upper = flatten_dbl(dat[[3]]),
           coef_pvalue = flatten_dbl(dat[[4]]),
           quad = rep(quadrant,
                      length(flatten(dat[[1]])))
           )
}

#flatten dat_quad
df_coef <- map_dfr(seq(4), dfs)
#remove ridge from name
```

```r
df_coef$name <- gsub("ridge\\(|\\)", "", df_coef$name)

# Figure 2
df_coef %>%
    select(-starts_with("HR_CI")) %>%
    filter(!between(HR, .99, 1.01)) %>%
    mutate(coef_pvalue = if_else(near(coef_pvalue, 0),
                                 coef_pvalue+0.1^17,
                                 coef_pvalue)) %>%
    ggplot(aes(x = log2(HR),
               y = -log10(coef_pvalue),
               colour = as.factor(quad))) +
          labs(colour = "Quadrant",
               x = 'log2 Hazard Ratio',
               y = '-log10 p-value') +
          geom_point(alpha = 0.75,
                     size = 1,
                     stroke = 1) +
          guides(colour = guide_legend(override.aes = list(alpha = 1))) +
          geom_text(aes(label=name),
                    alpha = 0.75,
                    vjust = 1.2,
                    show.legend = FALSE,
                    check_overlap = TRUE) +
          theme_minimal() +
          theme(plot.margin = margin(t = -15))
```




```
ggsave(here("img/2-volcano2.pdf"))
```

```
## Saving 6.5 x 4.5 in image
```

```
ggsave(here("img/2-volcano2.png"))
```

```
## Saving 6.5 x 4.5 in image
```

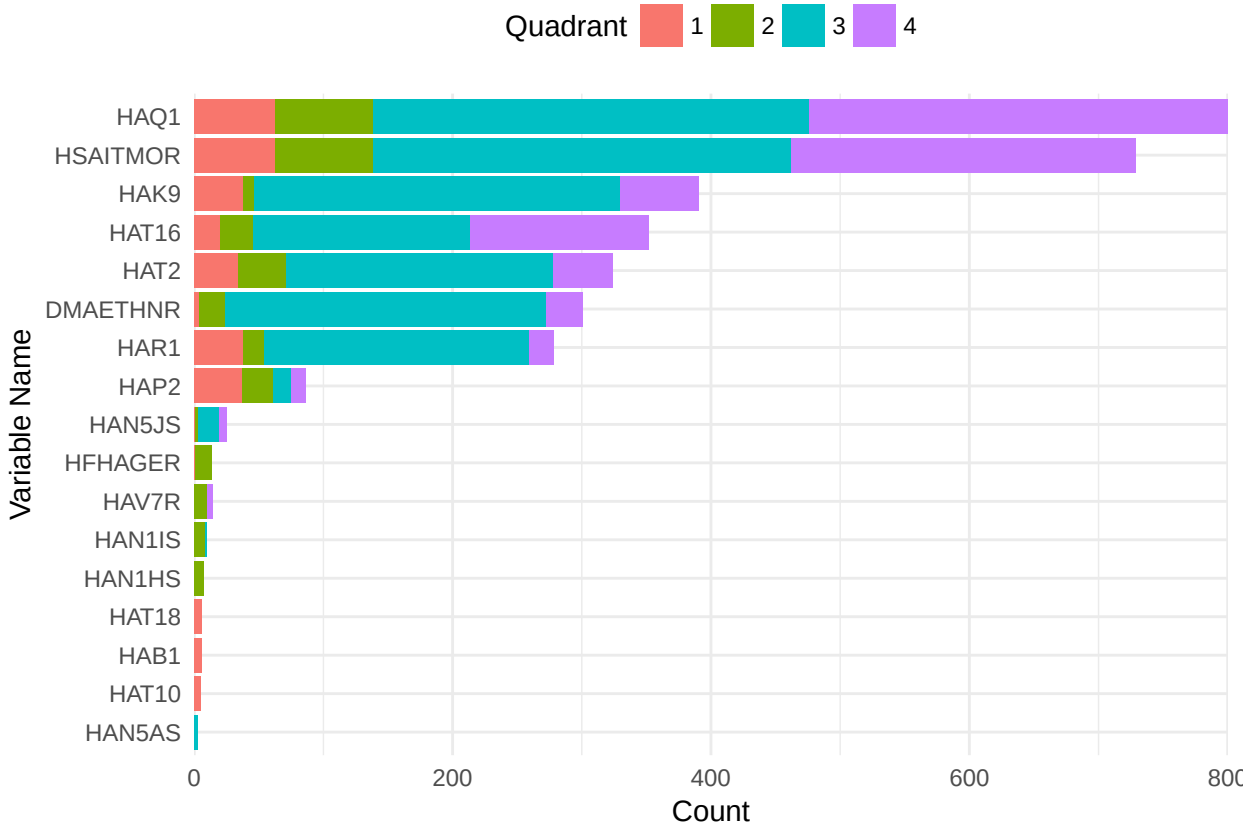
```
#filter out p-values greater than .1^10
df_sig <- df_coef %>%
    select(-starts_with("HR_CI")) %>%
    filter(coef_pvalue<.1^10)

#obtain the order by count for name
ord <- df_sig %>%
    count(name) %>%
    arrange(n) %>%
    select(name)

#create name factor variable with levels ordered by count
df_sig$ord_name <- factor(df_sig$name, levels=ord$name)

# Figure 3
df_sig %>%
    mutate_if(is.integer, as.factor) %>%
    ggplot(aes(ord_name,fill=quad)) +
    geom_bar(position = position_stack(reverse = TRUE)) +
```

```
    scale_y_continuous(expand = c(0,0)) +
    coord_flip() +
                  theme_minimal() +
    theme(legend.position = "top") +
                  labs(fill = "Quadrant",
                       x = 'Variable Name',
                       y = 'Count')
```




```
ggsave(here("img/3-varbar2.pdf"))
```

```
## Saving 6.5 x 4.5 in image
```

```
ggsave(here("img/3-varbar2.png"))
```

```
## Saving 6.5 x 4.5 in image
```

```
# Table 1
df_sig %>%
    group_by(quad) %>%
    rename(Name = name) %>%
    summarise(n = n()) %>%
    arrange(desc(n)) %>%
    knitr::kable()
```

quad	n
3	1807
4	904
2	324

quad	n
1	316

```
# Table 2
df_sig %>%
    group_by(name) %>%
    rename(Name = name) %>%
    summarise(medianHR = median(HR),
              n = n()) %>%
    arrange(desc(n)) %>%
    knitr::kable()
```

Name	medianHR	n
HAQ1	1.0732088	800
HSAITMOR	1.0004640	729
HAK9	1.1786575	391
HAT16	1.5340943	352
HAT2	1.5218932	324
DMAETHNR	1.0886708	301
HAR1	0.6683934	278
HAP2	0.7585256	86
HAN5JS	0.9977995	25
HAV7R	1.0001930	14
HFHAGER	1.0049141	14
HAN1IS	0.9983149	10
HAN1HS	0.9974929	7
HAB1	1.0000008	6
HAT18	1.0000018	6
HAT10	1.3519484	5
HAN5AS	0.9878849	3