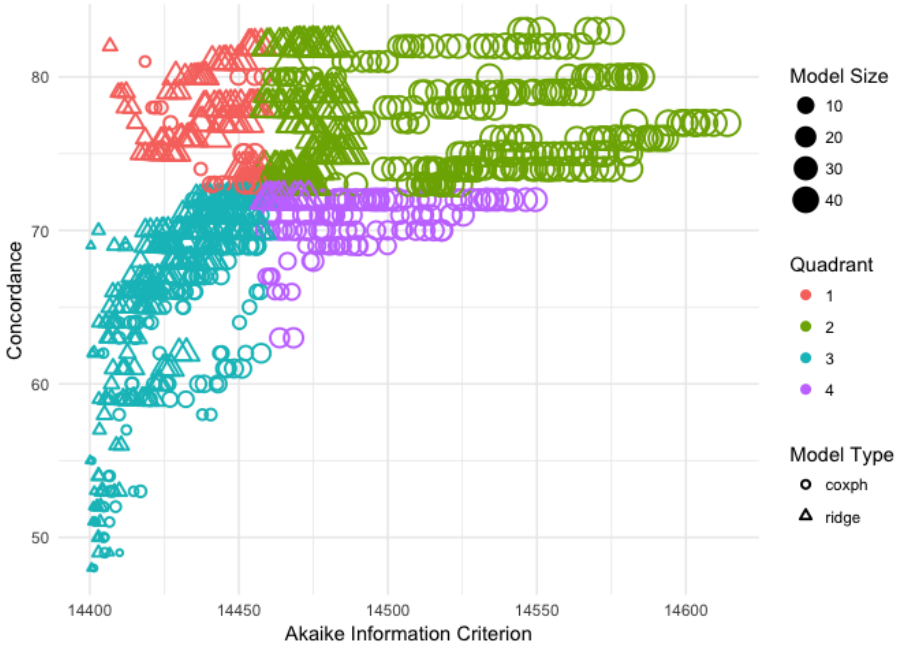
```
library(readr)
library(here)

## 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/6-model-diff-sizes.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,
```

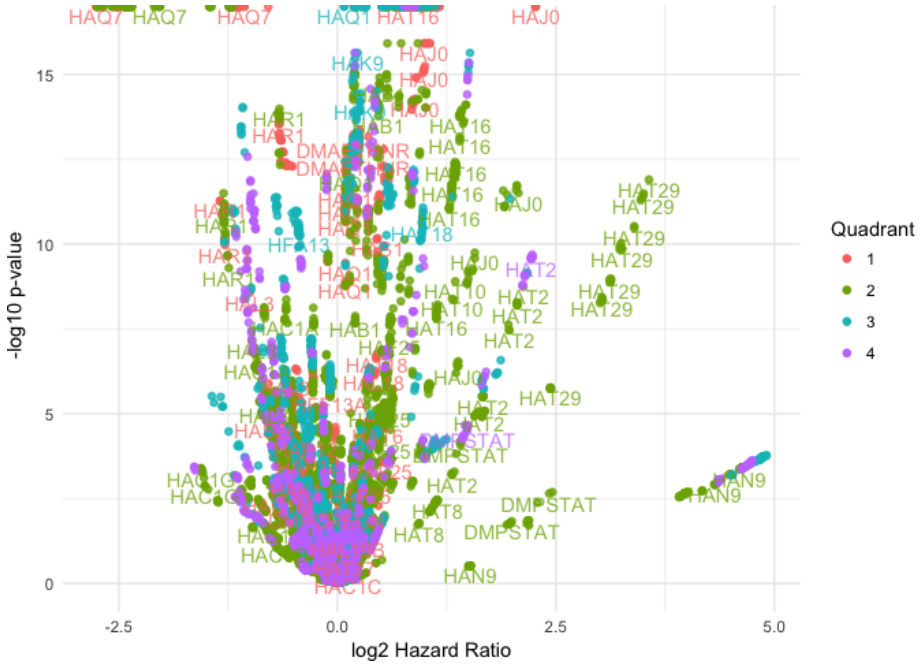
```
                        con <= median(con) &
                        aic > median(aic) ~ 4
                       )
                    )
          )

table(dat_quad$quad)

##
##   1   2   3   4
## 111 349 369 131

table(dat_quad$type)

##
## coxph ridge
##   480   480

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

## # A tibble: 8 x 3
## # Groups:   type [?]
##   type  quad      n
##   <chr> <fct> <int>
## 1 coxph 1        21
## 2 coxph 2       232
## 3 coxph 3       118
## 4 coxph 4       109
## 5 ridge 1        90
## 6 ridge 2       117
## 7 ridge 3       251
## 8 ridge 4        22

# 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")
```



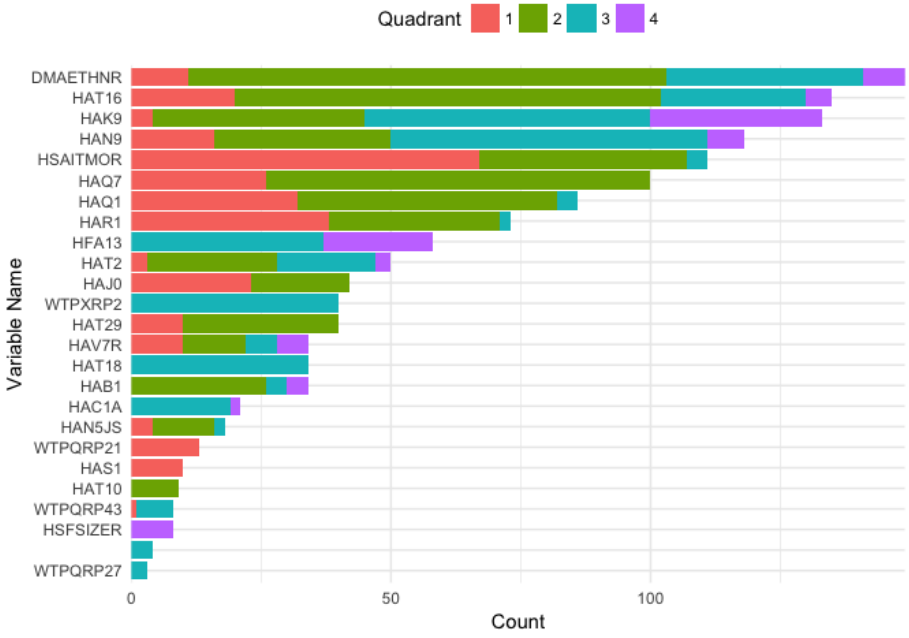

```
ggsave(here("img/1-quad.pdf"))

## Saving 7 x 5 in image

ggsave(here("img/1-quad.png"))

## Saving 7 x 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
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-volcano.pdf"))

## Saving 7 x 5 in image

ggsave(here("img/2-volcano.png"))

## Saving 7 x 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-varbar.pdf"))

## Saving 7 x 5 in image

ggsave(here("img/3-varbar.png"))

## Saving 7 x 5 in image

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

```
    knitr::kable()
```

quad

n

2

579

3

367

1

288

4

97

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

Name

medianHR

n

DMAETHNR

1.1566884

149

HAT16

1.7560810

135

HAK9

1.2189889

133

HAN9

1.7947078

118

HSAITMOR

1.0004575

111

HAQ7

0.3036946

100

HAQ1

1.0664794

86

HAR1

0.6274918

73

HFA13

0.6099651

58

HAT2

1.6533201

50

HAJ0

1.9863574

42

HAT29

2.0971949

40

WTPXRP2

0.9999864

40

HAB1

1.2485765

34

HAT18

1.5324396

34

HAV7R

1.0002031

34

HAC1A

0.6096885

21

HAN5JS

0.9980961

18

WTPQRP21

0.9999893

13

HAS1

1.3772235

10

HAT10

1.3949075

9

HSFSIZER

0.9185453

8

WTPQRP43

0.9999858

8

1.0260494

4

WTPQRP27

0.9999872

3