Coefficient Plot

Martin Stelmach

Coefficient Plot of Wine Types

```r
library(ggplot2)
library(directlabels)
library(broom)
library(broom.mixed)
```

```
## Registered S3 methods overwritten by 'broom.mixed':
##   method         from 
##   augment.lme    broom
##   augment.merMod broom
##   glance.lme     broom
##   glance.merMod  broom
##   glance.stanreg broom
##   tidy.brmsfit   broom
##   tidy.gamlss    broom
##   tidy.lme       broom
##   tidy.merMod    broom
##   tidy.rjags     broom
##   tidy.stanfit   broom
##   tidy.stanreg   broom
```

```
## 
## Attaching package: 'broom.mixed'
```

```
## The following object is masked from 'package:broom':
## 
##     tidyMCMC
```

```r
library(dotwhisker)
library(ggstance)
```

```
## 
## Attaching package: 'ggstance'
```

```
## The following objects are masked from 'package:ggplot2':
## 
##     geom_errorbarh, GeomErrorbarh
```

```r
library(tidyverse)
```

```
## -- Attaching packages ----------------------------------------------- tidyverse 1.2.1 --
```

```
## v tibble  2.1.3     v purrr   0.3.2
## v tidyr   1.0.0     v dplyr   0.8.3
## v readr   1.3.1     v stringr 1.4.0
## v tibble  2.1.3     v forcats 0.4.0
```

```
## -- Conflicts -------------------------------------------------- tidyverse_conflicts() --
## x dplyr::filter()            masks stats::filter()
## x ggstance::geom_errorbarh() masks ggplot2::geom_errorbarh()
## x dplyr::lag()               masks stats::lag()
## x broom.mixed::tidyMCMC()    masks broom::tidyMCMC()
```

```
library(dplyr)
library(rattle)
```

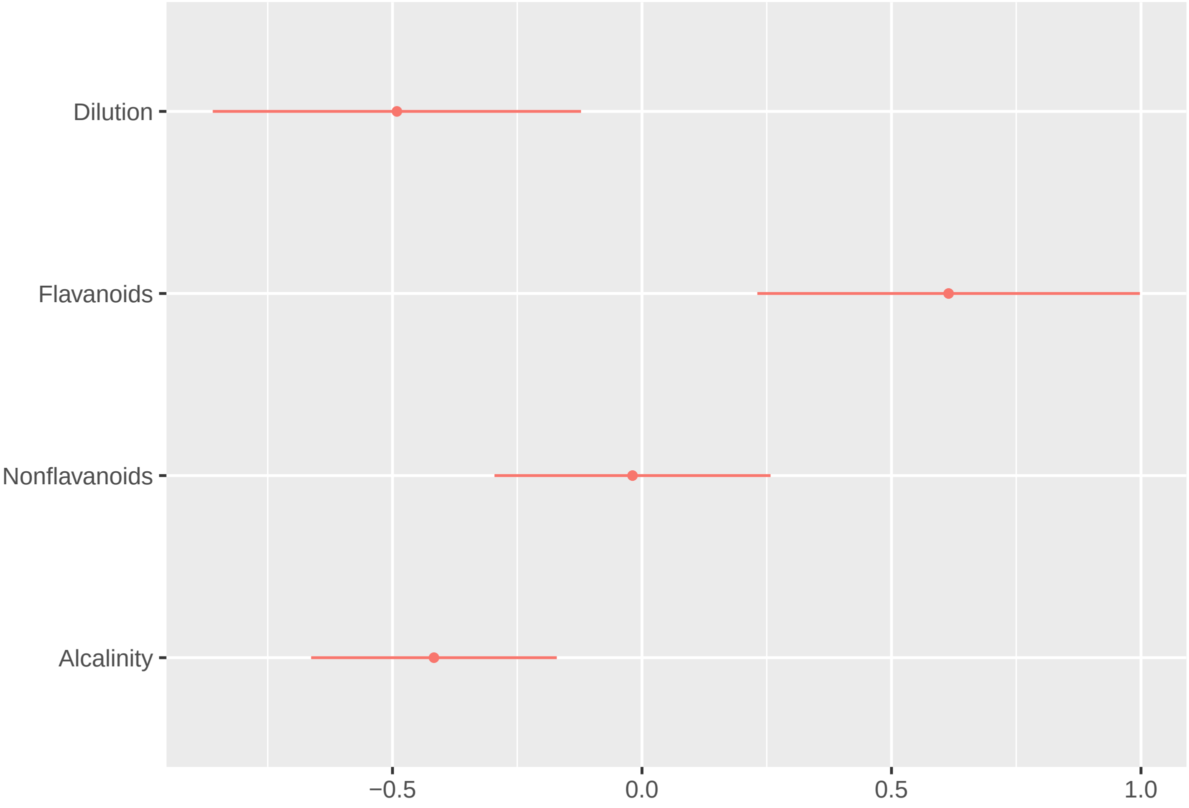
```
## Rattle: A free graphical interface for data science with R.
## Version 5.2.0 Copyright (c) 2006-2018 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```
data("wine")
```

Creating the Graph

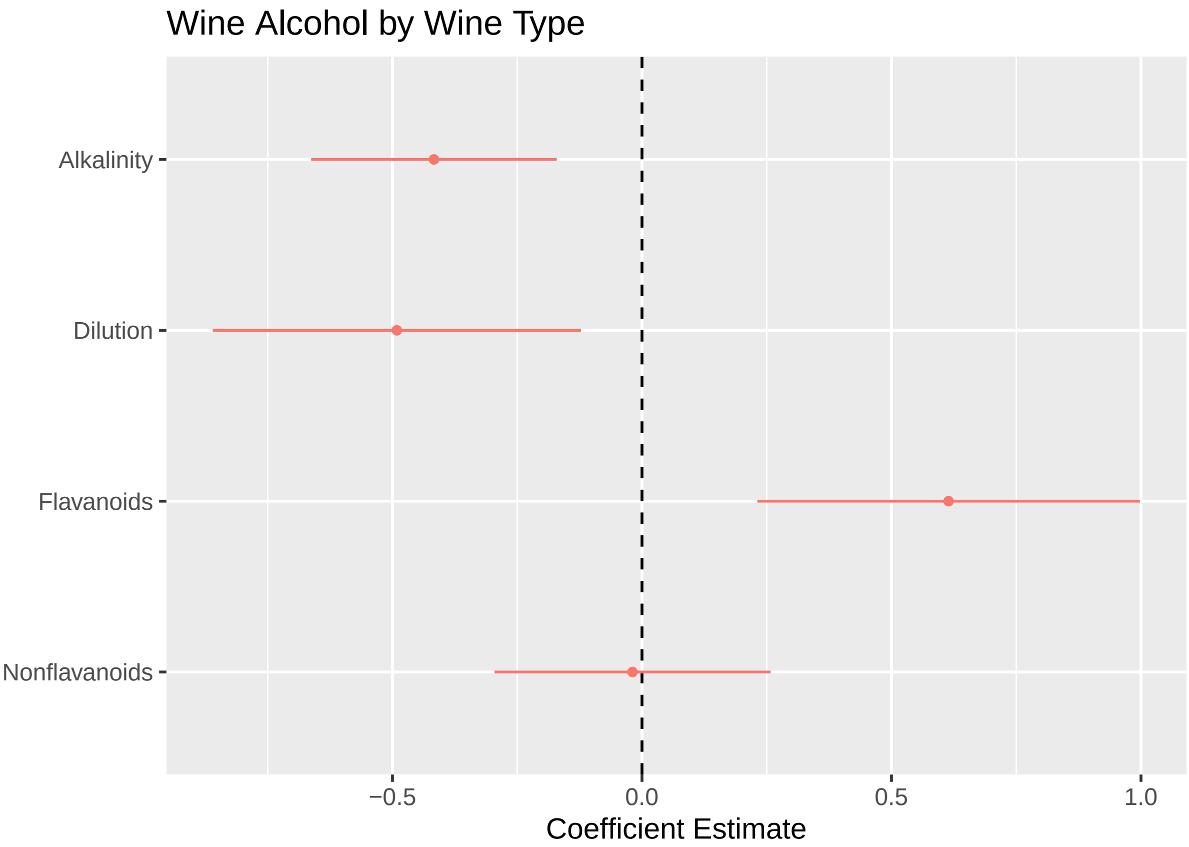
I began by sorting the data into a linear model with Alcohol being a response variable for serveral other factors and I made a simple dot and whisker plot of the model.

```
#we want type, alcohol, dilution, flavanoids, nonflavanoids, maybe alcalinity as well
gg0 <- lm(Alcohol ~ Dilution + Flavanoids + Nonflavanoids + Alcalinity, data = wine)
dwplot(gg0)
```



Improvements

When first creating the graph I realized that many things such as title and axis and all the things that make graphs look nice.



This looks a lot nicer. Has things like labels and correct variable names. Now I want to do this but for several types of wines. So in order to do this we will use the dplyr package to seperate the wine by type and use them in the same model as comparisons to one another.

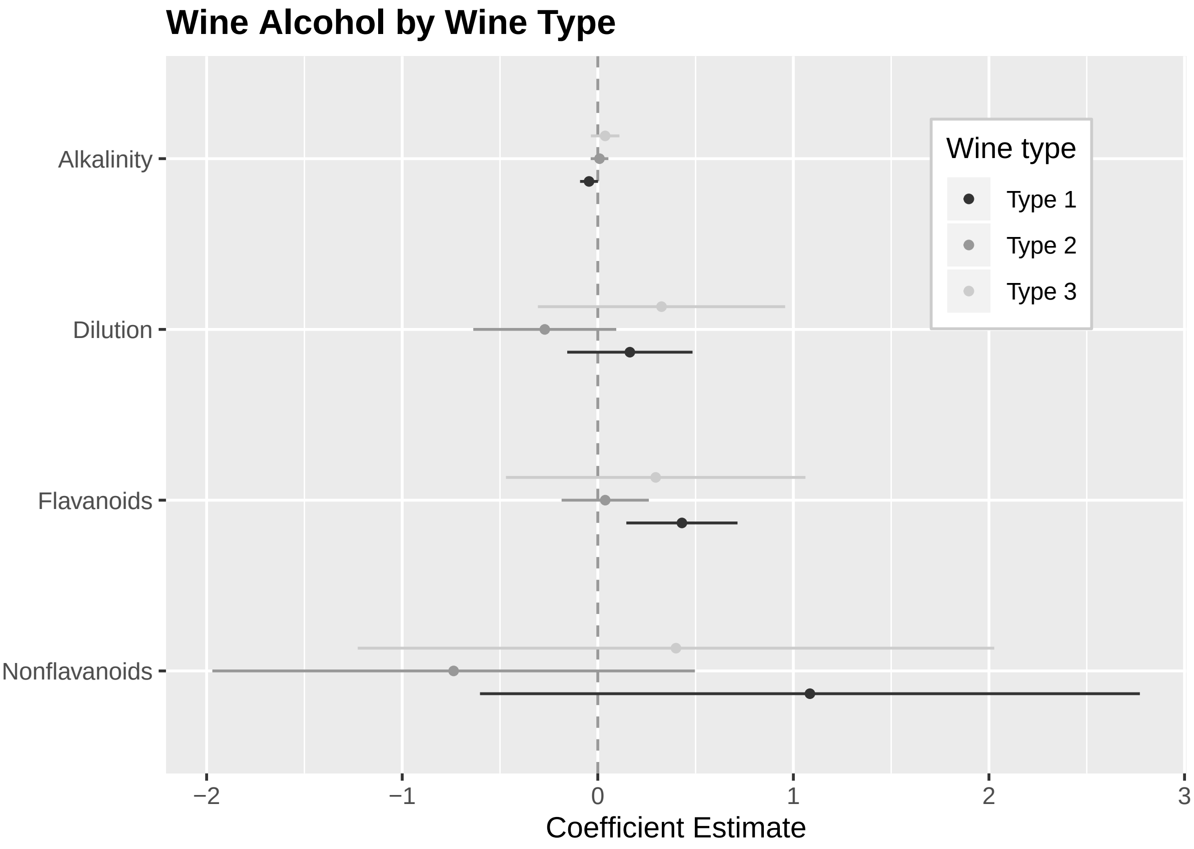
```
by_type <- wine %>%
  group_by(Type) %>%
  do(tidy(lm(Alcohol ~ Dilution + Flavanoids + Nonflavanoids + Alcalinity, data = .))) %>%
  rename(model=Type) %>%
  relabel_predictors(Alcalinity = "Alkalinity")

by_type
```

```
## # A tibble: 15 x 6
## # Groups:   model [3]
##    model term          estimate std.error statistic  p.value
##    <fct> <fct>            <dbl>     <dbl>     <dbl>    <dbl>
##  1 1     Alkalinity    -0.0448     0.0235    -1.91  6.15e- 2
##  2 1     (Intercept)   12.4        0.947     13.1   2.23e-18
##  3 1     Dilution       0.164      0.163      1.00  3.20e- 1
##  4 1     Flavanoids     0.430      0.145      2.97  4.48e- 3
##  5 1     Nonflavanoids  1.08       0.861      1.26  2.13e- 1
##  6 2     Alkalinity     0.00881    0.0231     0.381 7.04e- 1
##  7 2     (Intercept)   13.0        0.545     24.0   2.89e-34
##  8 2     Dilution      -0.271      0.186     -1.45  1.51e- 1
##  9 2     Flavanoids     0.0379     0.114      0.333 7.40e- 1
## 10 2     Nonflavanoids -0.737      0.629     -1.17  2.46e- 1
## 11 3     Alkalinity     0.0376     0.0372     1.01  3.18e- 1
## 12 3     (Intercept)   11.4        0.982     11.6   8.04e-15
## 13 3     Dilution       0.326      0.322      1.01  3.18e- 1
## 14 3     Flavanoids     0.296      0.391      0.758 4.53e- 1
## 15 3     Nonflavanoids  0.400      0.830      0.481 6.33e- 1
```

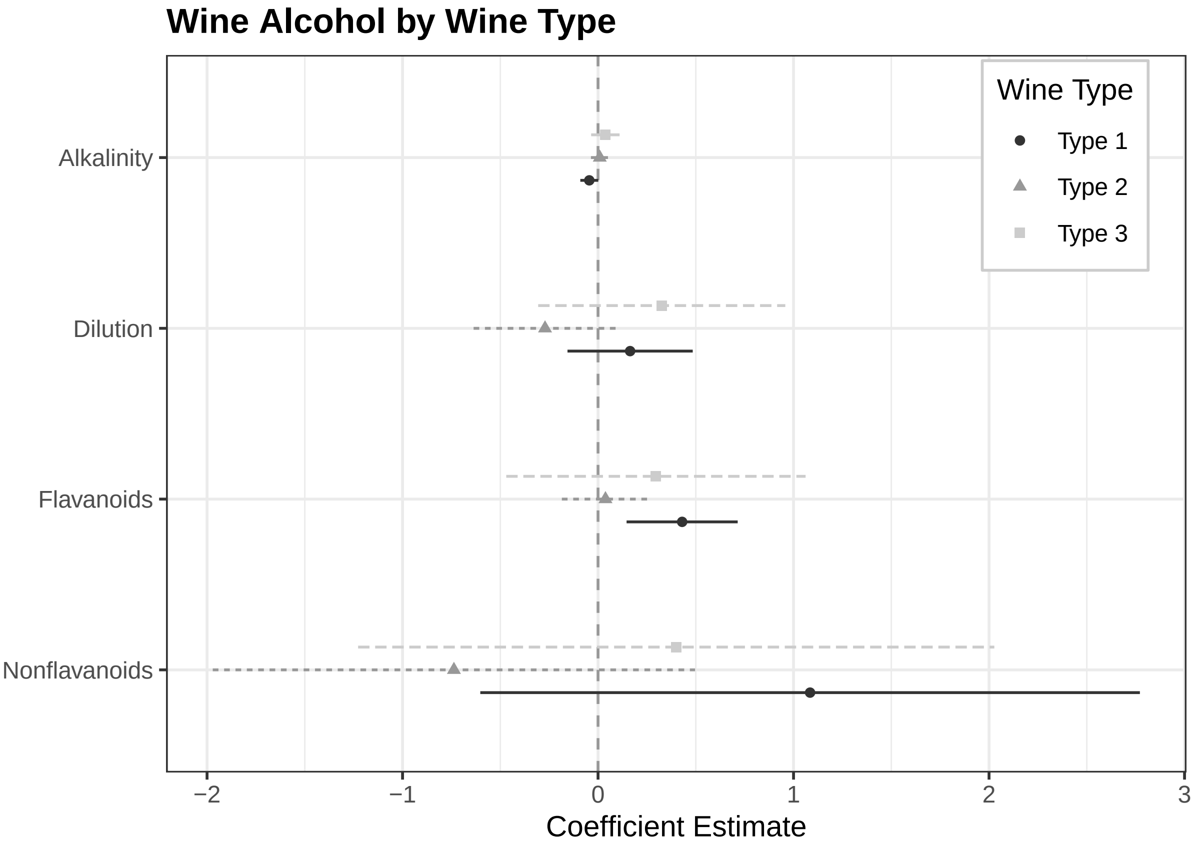
This works successfully and now we can plot the model by each respective type of wine. We use the gray scale to make it visible for everyone. Although by colour would probably be the best way to do this graph, I want to see if I can make one that stands out in a gray scale. With some research on the internet, colour gray60 and gray80 are the best for grid patterns and vertical lines.

```
dwplot(by_type,
       vline = geom_vline(xintercept = 0, colour = "grey60", linetype = 2)) +
  xlab("Coefficient Estimate") + ylab("") +
  ggtitle("Wine Alcohol by Wine Type") +
  theme(plot.title = element_text(face="bold"),
        legend.position = c(0.75, 0.62),
        legend.justification = c(0, 0),
        legend.background = element_rect(colour="grey80"),
        legend.title.align = .5) +
  scale_colour_grey(name = "Wine type",
                    breaks = c( 1, 2, 3),
                    labels = c("Type 1", "Type 2", "Type 3"))
```



This is the same graph with a monochromatic grey scale so that they are viewable for everyone as well as shapes to differentiate the different varieties of wine. I like these graphs and I think they are appropriate for the model.

```
#So jonathan can see
dwplot(by_type,
       vline = geom_vline(xintercept = 0, colour = "grey60",linetype = 2),
       dot_args = list(aes(shape = model)),
       whisker_args = list(aes(linetype = model)))+
  xlab("Coefficient Estimate") + ylab("") +
  theme_bw()+
  ggtitle("Wine Alcohol by Wine Type") +
  theme(plot.title = element_text(face="bold"),
        legend.position = c(0.8, 0.7),
        legend.justification = c(0, 0),
        legend.background = element_rect(colour="grey80"),
        legend.title.align = .5) +
  scale_colour_grey(name = "Wine Type",
                    breaks = c(1, 2, 3),
                    labels = c("Type 1", "Type 2", "Type 3")) +
  scale_shape_discrete(name = "Wine Type",
              breaks = c(1,2,3),
              labels = c("Type 1", "Type 2", "Type 3"))
```



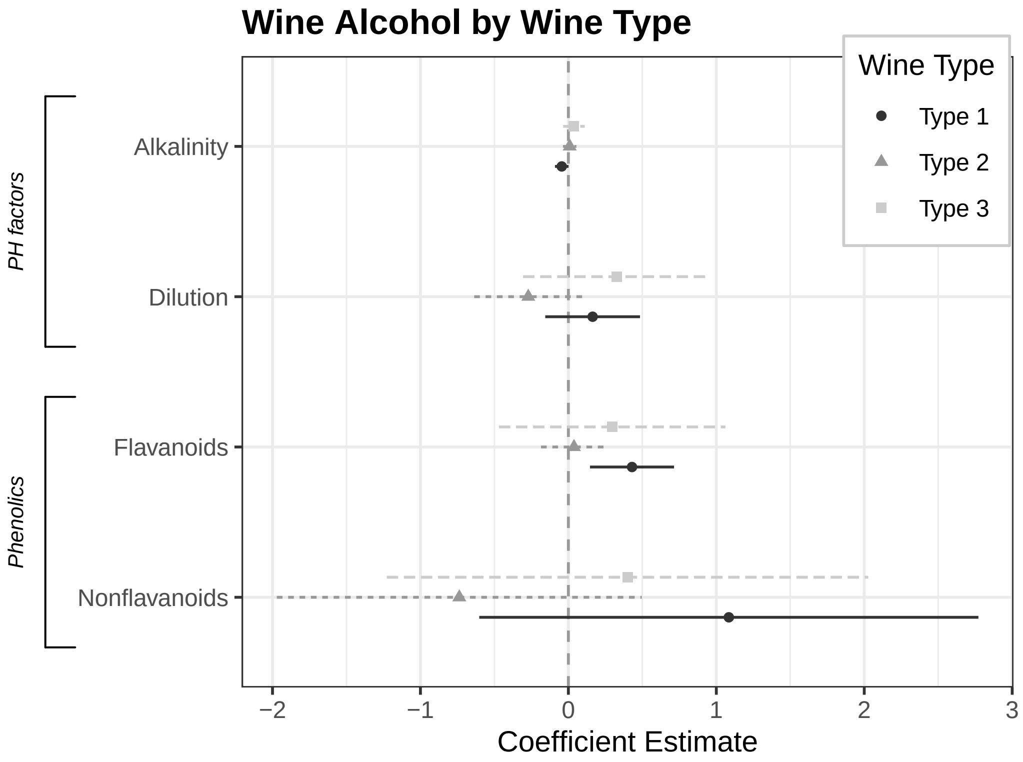
Each line is also using different line types which I found particularly nice. Easier to differentiate of a black and white scale. If it was a colour scale I would have used solid colours but I think that it helps with the black and white scale. An arguement could be made that the Alkailinty is difficult to view, to which I agree and obviously the solution is to use a colour scale however I wanted to challenge myself and try and find solutions to this problem, hence the use of shapes and line types and the use of a white background. I tried to make a the x axis something different but they are all different units or measurement and I did not really know how to normalize these into a single scale. This being said we can show there is a divide in the two groups

```
brackets <- list(c("Phenolics", "Flavanoids", "Nonflavanoids"),
                 c("PH factors", "Alkalinity", "Dilution"))
```

These brackets are how I will be dividing the classes of variables.

Final Product

```
{dwplot(by_type,
       vline = geom_vline(xintercept = 0, colour = "grey60",linetype = 2),
       dot_args = list(aes(shape = model)),
       whisker_args = list(aes(linetype = model)))+
  xlab("Coefficient Estimate") + ylab("") +
  theme_bw()+
  ggtitle("Wine Alcohol by Wine Type") +
  theme(plot.title = element_text(face="bold"),
        legend.position = c(0.78, 0.7),
        legend.justification = c(0, 0),
        legend.background = element_rect(colour="grey80"),
        legend.title.align = .5) +
  scale_colour_grey(name = "Wine Type",
                    breaks = c(1, 2, 3),
                    labels = c("Type 1", "Type 2", "Type 3")) +
  scale_shape_discrete(name = "Wine Type",
              breaks = c(1,2,3),
              labels = c("Type 1", "Type 2", "Type 3"))} %>%
  add_brackets(brackets)
```



I notice all my graphs are different sizes hence my legends are in different places. I dont know how to change that but that would be something I would work out in the future.