normalize

Alison Gale

2/27/2021

Testing strategies for normalizing coefficients

Start by loading the six month data:

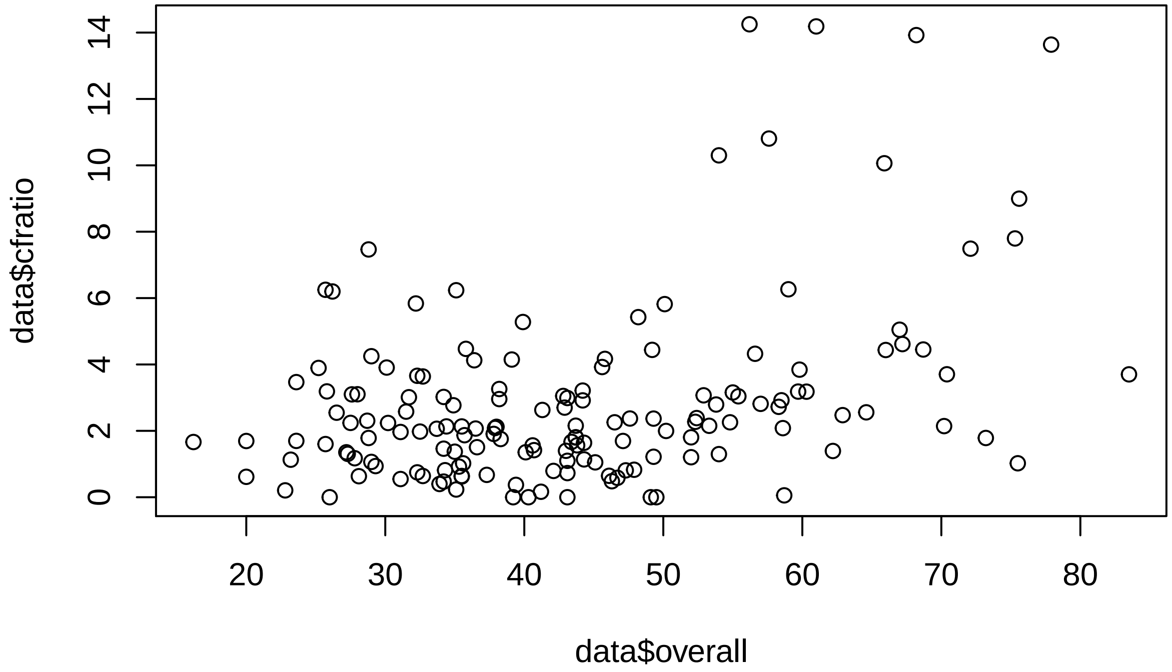
```
data <- read.csv(file = '../prepped_data/six_month_outlier_screened.csv')
data <- data[!(is.na(data$gdp_pc) | is.na(data$democracy_index)),]
```

As a baseline, run a regression on the overall GHSI score:

```
old_fit = lm(formula = cfratio ~ overall, data = data)
summary(old_fit)
```

```
##
## Call:
## lm(formula = cfratio ~ overall, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.3174 -1.4337 -0.6134  0.7258 10.4173
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.55833    0.65741  -0.849    0.397
## overall      0.07812    0.01437   5.437 2.09e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.501 on 154 degrees of freedom
## Multiple R-squared:  0.161,  Adjusted R-squared:  0.1556
## F-statistic: 29.56 on 1 and 154 DF,  p-value: 2.086e-07
```

```
plot(data$overall, data$cfratio)
```



First we run the following regression to figure out which of the original sub-components are large contributors:

```
summary(lm(formula = cfratio ~ prev_emergence_pathogens + early_detection + rapid_response + robust_heal
```

```
##
## Call:
## lm(formula = cfratio ~ prev_emergence_pathogens + early_detection +
##     rapid_response + robust_health_sector + commitments + risk_environment,
##     data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.4459 -1.4882 -0.5287  0.6906 10.5374
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -0.606666   1.200418  -0.505    0.614
## prev_emergence_pathogens  0.033176   0.026468   1.253    0.212
## early_detection           0.004320   0.014264   0.303    0.762
## rapid_response            0.003816   0.022521   0.169    0.866
## robust_health_sector      0.018860   0.026781   0.704    0.482
## commitments               0.034144   0.022441   1.522    0.130
## risk_environment         -0.008933   0.019798  -0.451    0.652
##
## Residual standard error: 2.518 on 149 degrees of freedom
## Multiple R-squared:  0.1775, Adjusted R-squared:  0.1444
## F-statistic: 5.361 on 6 and 149 DF,  p-value: 4.878e-05
```

Suppose we run the following regression to get coefficients for our model:

```
fit = lm(formula = cfratio ~ prev_emergence_pathogens + robust_health_sector + commitments + is_island
summary(fit)
```

```
##
## Call:
## lm(formula = cfratio ~ prev_emergence_pathogens + robust_health_sector +
##     commitments + is_island + gdp_pc + democracy_index, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.9317 -1.4614 -0.5310  0.6803 10.3025
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -1.135e+00  9.739e-01  -1.165   0.2458
## prev_emergence_pathogens  2.705e-02  2.445e-02   1.106   0.2704
## robust_health_sector      1.118e-02  2.376e-02   0.471   0.6386
## commitments               3.688e-02  2.107e-02   1.751   0.0821 .
## is_islandTRUE            -6.497e-01  6.083e-01  -1.068   0.2872
## gdp_pc                   -5.026e-07  1.378e-05  -0.036   0.9709
## democracy_index           1.489e-01  1.316e-01   1.131   0.2599
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.505 on 149 degrees of freedom
## Multiple R-squared:  0.1857, Adjusted R-squared:  0.1529
## F-statistic: 5.661 on 6 and 149 DF,  p-value: 2.533e-05
```

Now we retrieve estimated values for the data:

```
estimates = fitted.values(fit)
```

And we normalize these values from 0 to 100:

```
norm = rescale(estimates, c(100, 0))
```

Add this back to the data frame:

```
data$new_overall = norm
```

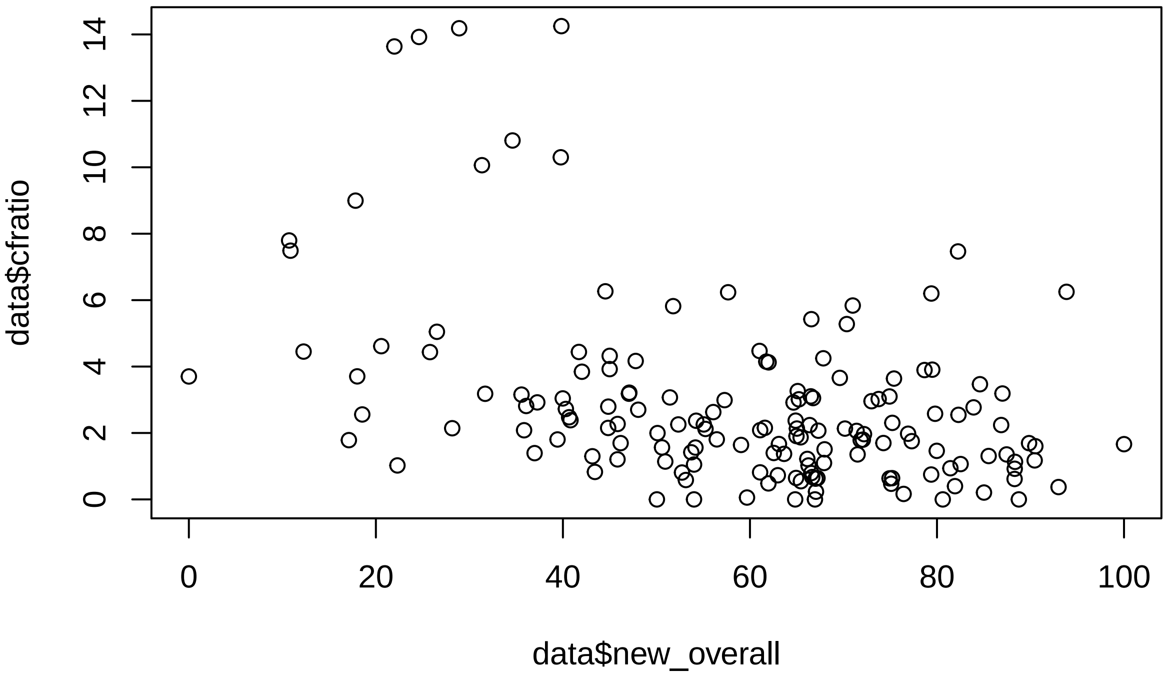
Finally, we run a new regression:

```
new_fit = lm(formula = cfratio ~ new_overall, data = data)
summary(new_fit)
```

```
##
## Call:
## lm(formula = cfratio ~ new_overall, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.9317 -1.4614 -0.5310  0.6803 10.3025
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.235813   0.605179  10.304  < 2e-16 ***
## new_overall -0.057462   0.009698  -5.925 1.97e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.464 on 154 degrees of freedom
## Multiple R-squared:  0.1857, Adjusted R-squared:  0.1804
## F-statistic: 35.11 on 1 and 154 DF,  p-value: 1.968e-08
```

Plot the data:

```
plot(data$new_overall, data$cfratio)
```



In summary, we can get the correlation to go in the correct direction with the new score and the plots of the data look a little better. The improvement was more noticeable for cases-per-capita. Additionally the statistical significance of the intercept is higher. We might be able to improve this by adding in our confounding variables (possibly to replace the sub-components with weights closer to zero).