Modeling

```
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(tidyr)
library(ggplot2)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
```

```
## v tibble  3.1.8     v stringr 1.5.0
## v readr   2.1.3     v forcats 0.5.2
## v purrr   1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(modelr)
```

```
setwd('..')
path <- getwd()
```

Using the U.S. Transgender Population Health Survey (TransPop) originally available from https://www.icpsr.umich.edu/web/ICPSR/studies/37938

```
file <- paste(path, 'TransPopData/37938-0001-Data.rda', sep='/')
```

```
load(file)
```

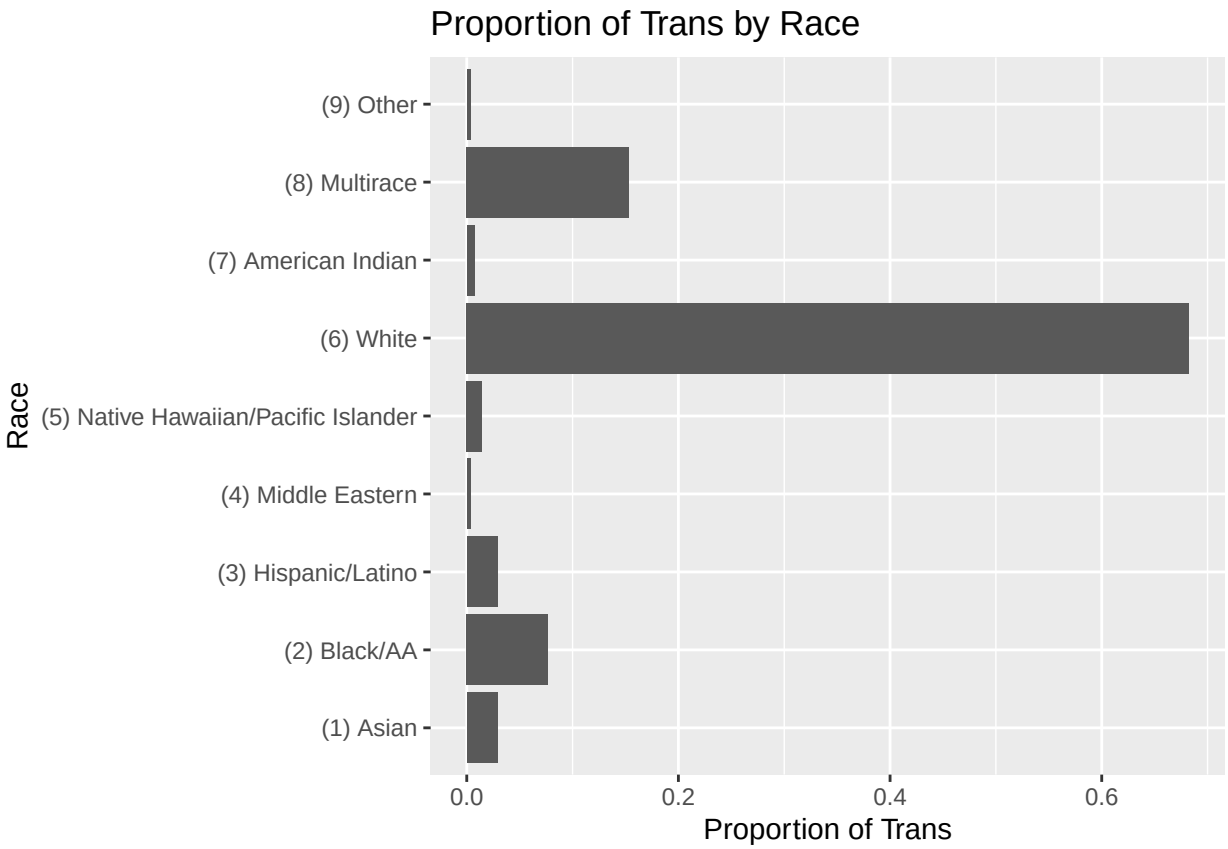
```
raw_data <- da37938.0001
```

Part A

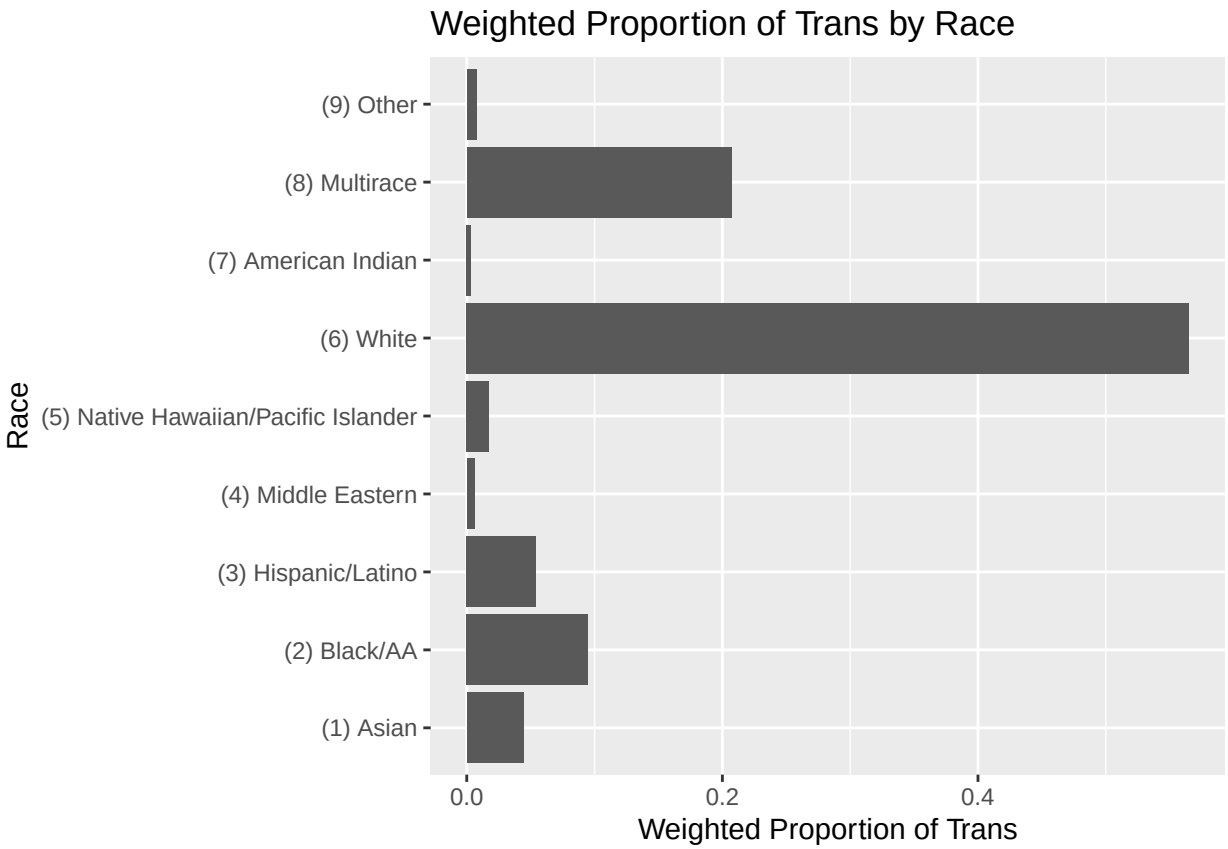
1. Comparing the weighted and unweighted distributions of trans people of different races and ethnicities:

```
data <- select(raw_data, c('STUDYID', 'WEIGHT', 'RACE', 'GENDER', 'SEXUALID'))
```

```
grouped <- data %>%
            group_by(RACE) %>% summarize(WEIGHT_GRP = sum(WEIGHT), COUNT = n())

grouped <- grouped %>% mutate(WEIGHTED_PROP = WEIGHT_GRP/sum(WEIGHT_GRP), PROP = COUNT/sum(COUNT))

ggplot(grouped, aes(x = RACE, y = PROP)) +
  geom_bar(stat = "identity", position=position_dodge()) +
  labs(title = "Proportion of Trans by Race",
       x = "Race",
       y = "Proportion of Trans") + coord_flip()
```




```
ggplot(grouped, aes(x = RACE, y = WEIGHTED_PROP)) +
  geom_bar(stat = "identity", position=position_dodge()) +
  labs(title = "Weighted Proportion of Trans by Race",
       x = "Race",
       y = "Weighted Proportion of Trans") + coord_flip()
```

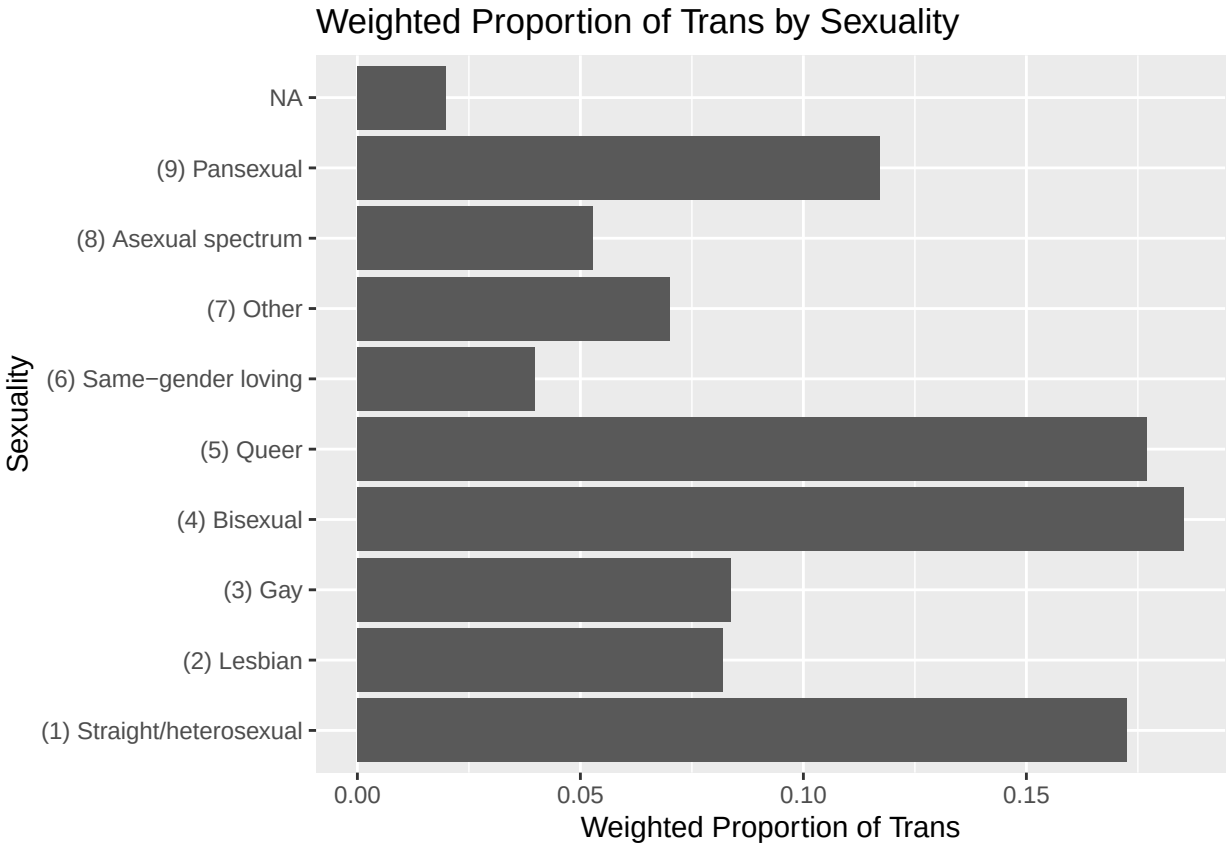


White ethnicity are over represented. Rest all fall in minority. Middle eastern are the most underrepresented (not considering other).

2. compare the weighted and unweighted distributions of trans people with different sexual orientations

```
grouped <- data %>%
            group_by(SEXUALID) %>% summarize(WEIGHT_GRP = sum(WEIGHT), COUNT = n())

grouped <- grouped %>% mutate(WEIGHTED_PROP = WEIGHT_GRP/sum(WEIGHT_GRP), PROP = COUNT/sum(COUNT))

ggplot(grouped, aes(x = SEXUALID, y = PROP)) +
  geom_bar(stat = "identity", position=position_dodge()) +
  labs(title = "Proportion of Trans by Sexuality",
       x = "Sexuality",
       y = "Proportion of Trans") + coord_flip()
```


```
ggplot(grouped, aes(x = SEXUALID, y = WEIGHTED_PROP)) +
  geom_bar(stat = "identity", position=position_dodge()) +
  labs(title = "Weighted Proportion of Trans by Sexuality",
       x = "Sexuality",
       y = "Weighted Proportion of Trans") + coord_flip()
```



Asexual and same gender loving are under represented (ignoring unknown (NA)).

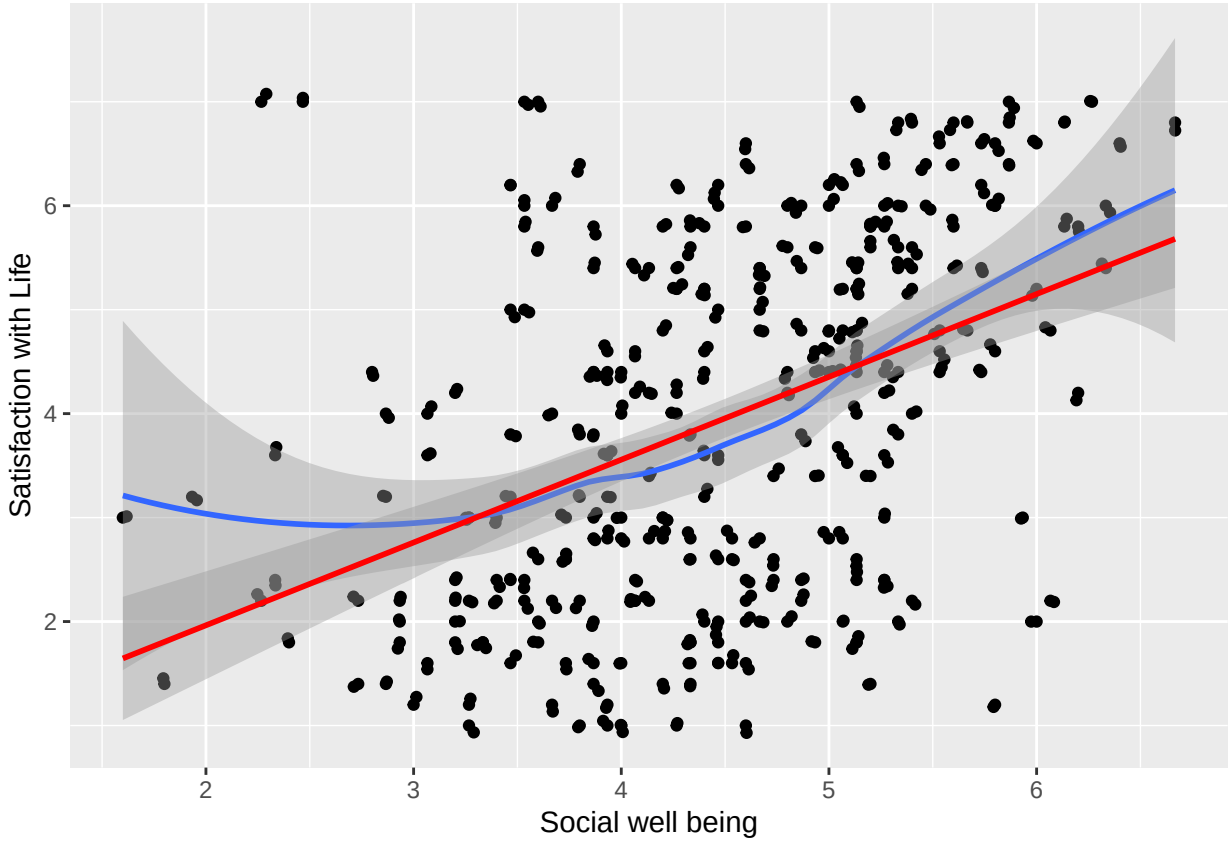
Part B

The survey includes several validated scales for measuring constructs related to identity, stress, and health. We would like to use these scales to build a model for predicting satisfaction with life among trans people.

```
filter_data <- select(raw_data, c('LIFESAT_I', 'SOCIALWB_I', 'NONAFFIRM_I', 'NONDISCLOSURE_I',  'HCTHREA
```

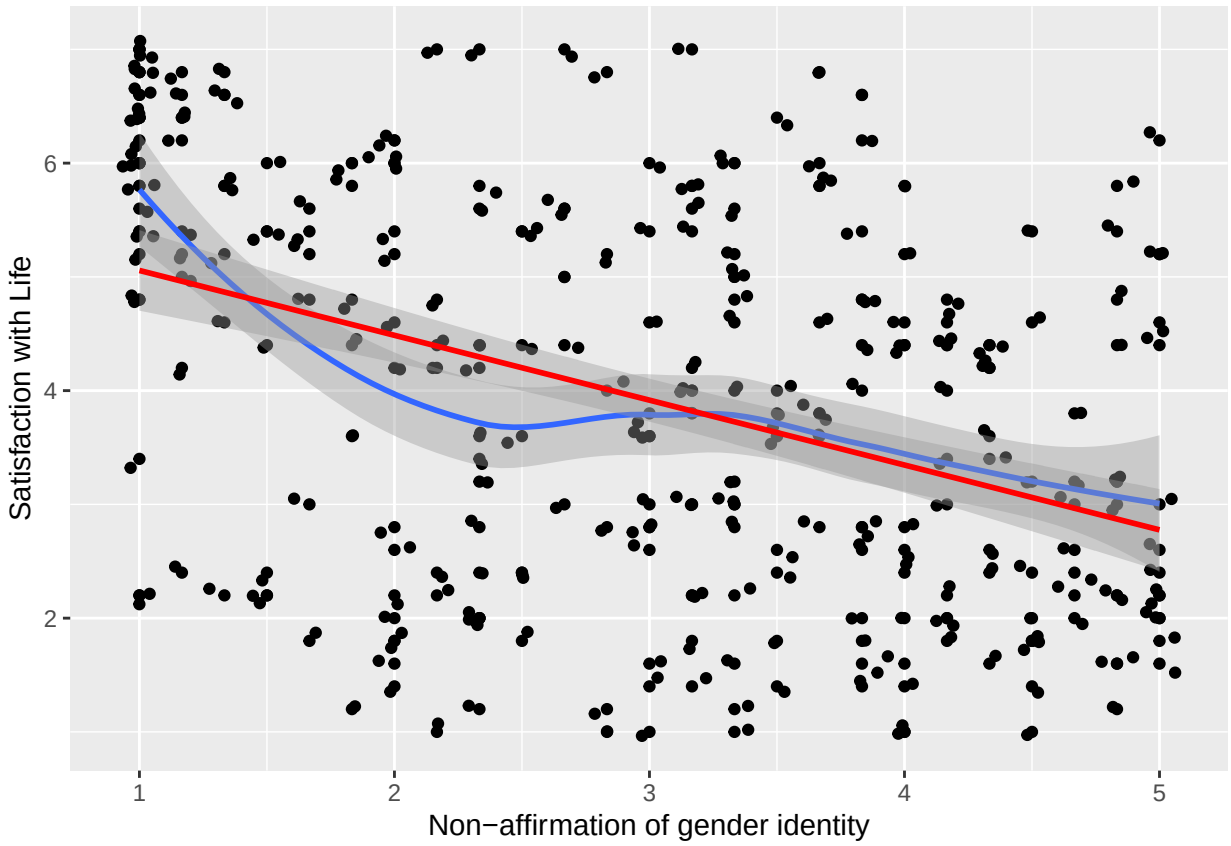
```
ggplot(filter_data, aes(x=SOCIALWB_I, y=LIFESAT_I)) +
  geom_point() +
  geom_jitter() +
  geom_smooth() +
  geom_smooth(method="lm", color='red') +
  labs(x='Social well being', y="Satisfaction with Life")
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```



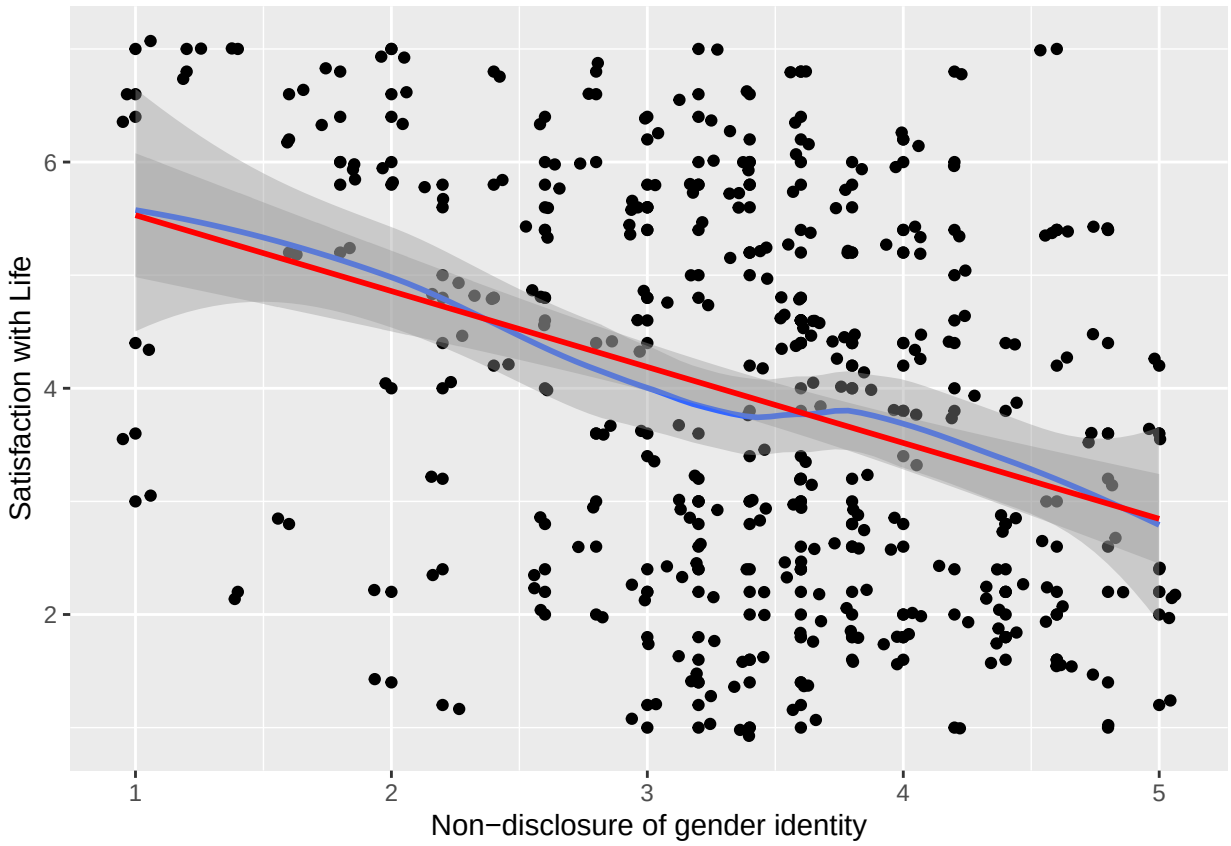

```
ggplot(filter_data, aes(x=NONAFFIRM_I, y=LIFESAT_I)) +
  geom_point() +
  geom_jitter() +
  geom_smooth() +
  geom_smooth(method="lm", color='red') +
  labs(x='Non-affirmation of gender identity', y="Satisfaction with Life")
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```



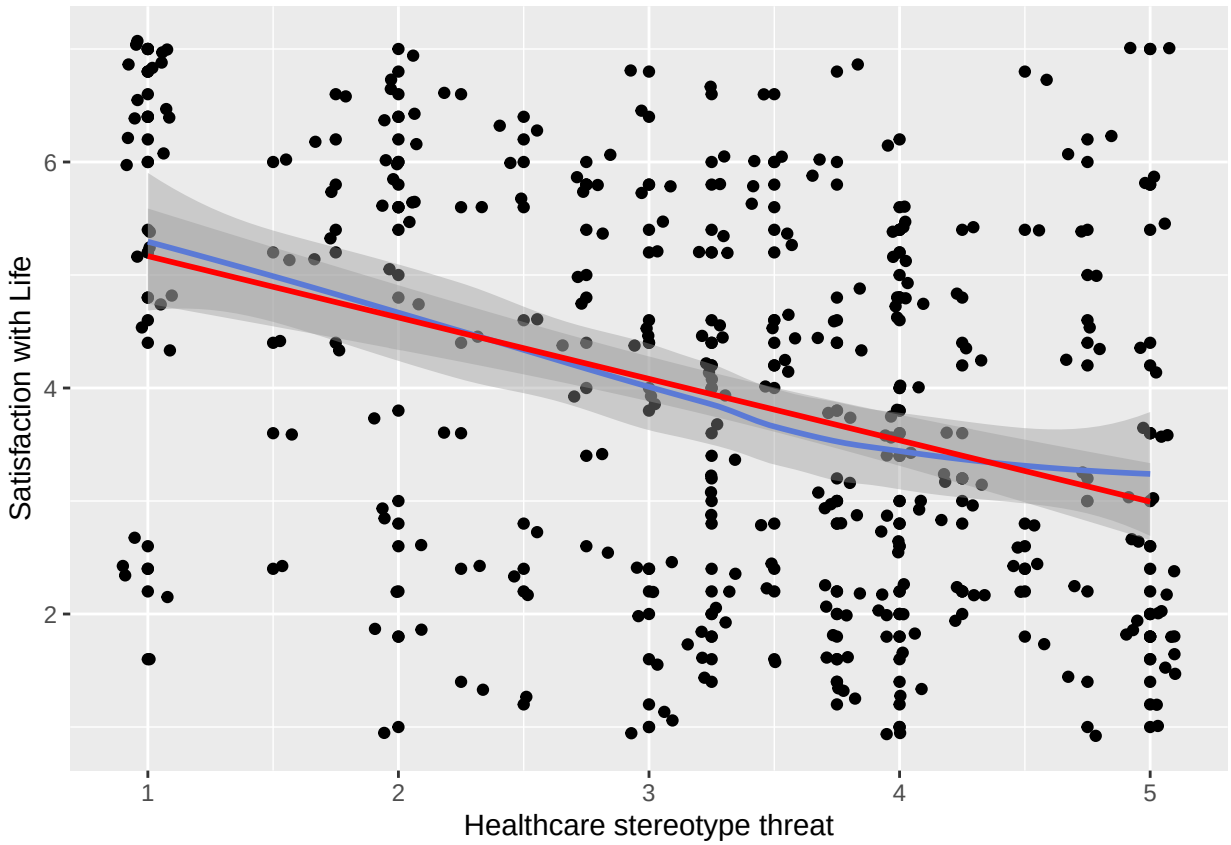

```
ggplot(filter_data, aes(x=NONDISCLOSURE_I, y=LIFESAT_I)) +
  geom_point() +
  geom_jitter() +
  geom_smooth() +
  geom_smooth(method="lm", color='red') +
  labs(x='Non-disclosure of gender identity', y="Satisfaction with Life")
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```



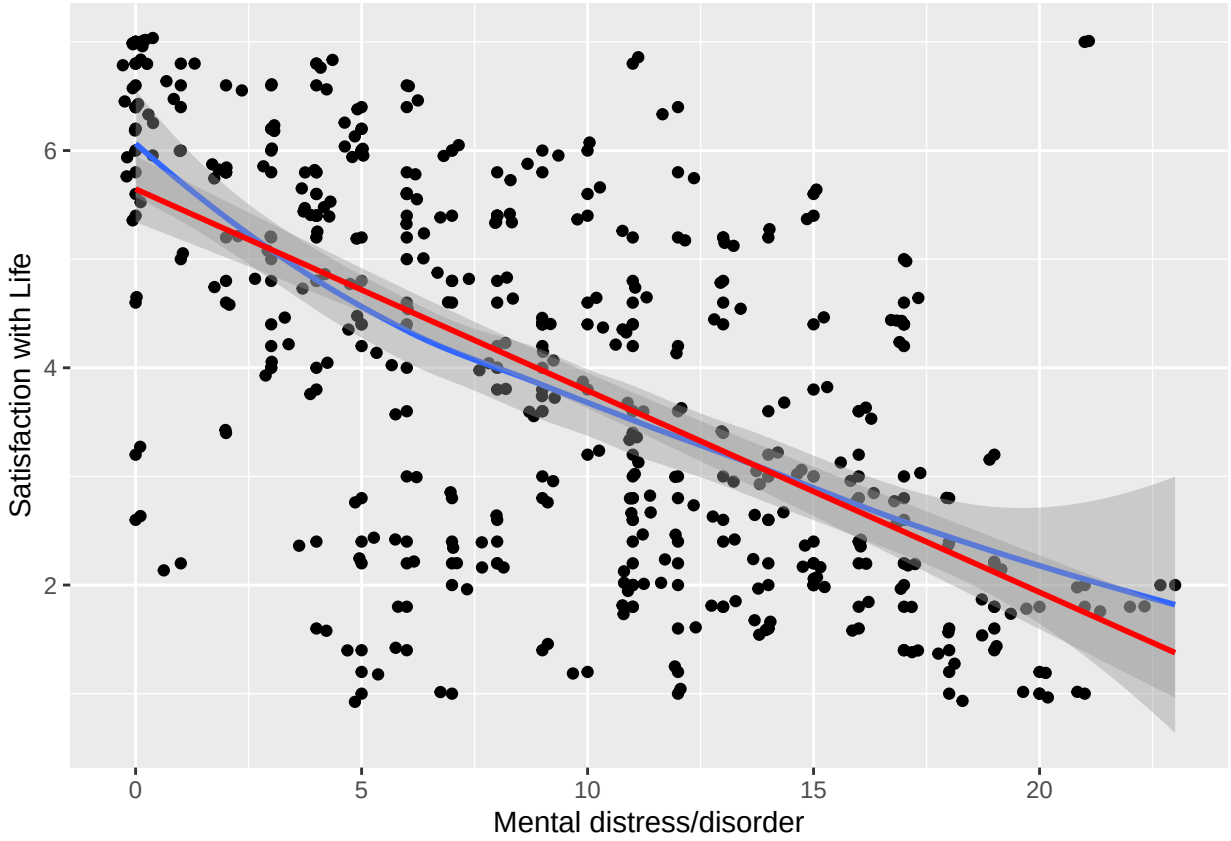

```
ggplot(filter_data, aes(x=HCTHREAT_I, y=LIFESAT_I)) +
  geom_point() +
  geom_jitter() +
  geom_smooth() +
  geom_smooth(method="lm", color='red') +
  labs(x='Healthcare stereotype threat', y="Satisfaction with Life")
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```



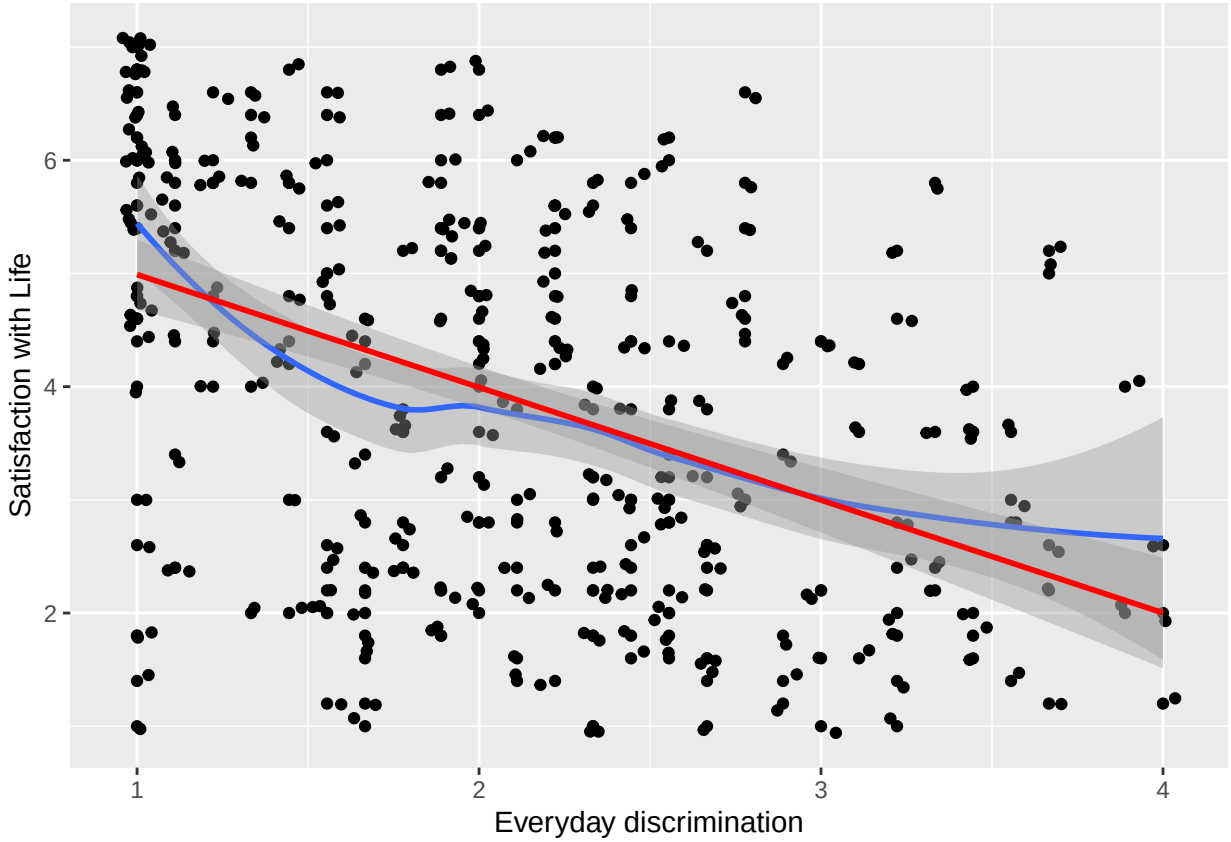

```
ggplot(filter_data, aes(x=KESSLER6_I, y=LIFESAT_I)) +
  geom_point() +
  geom_jitter() +
  geom_smooth() +
  geom_smooth(method="lm", color='red') +
  labs(x='Mental distress/disorder', y="Satisfaction with Life")
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```




```
ggplot(filter_data, aes(x=EVERYDAY_I, y=LIFESAT_I)) +
  geom_point() +
  geom_jitter() +
  geom_smooth() +
  geom_smooth(method="lm", color='red') +
  labs(x='Everyday discrimination', y="Satisfaction with Life")
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```



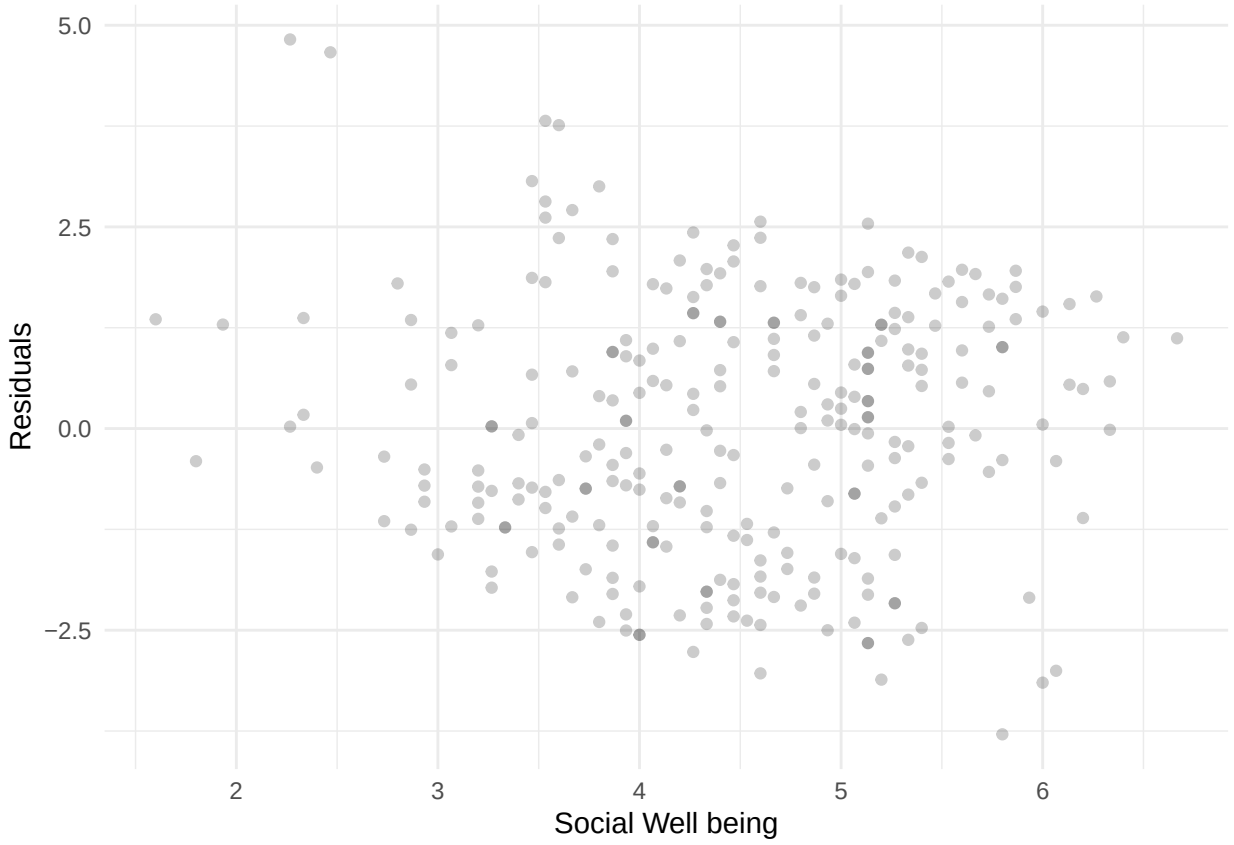
Social well being (SOCIALWB_I) has positive relationship with Satisfaction with life (LIFESAT_I) Mental distress/disorder (KESSLER6_I) has a negative relationship with Satisfaction with life (LIFESAT_I) All others have weak negative relationship with Satisfaction with life (LIFESAT_I)

Since Social well being seems to have the strongest relation with life satisfaction, I chose it as a predictor.

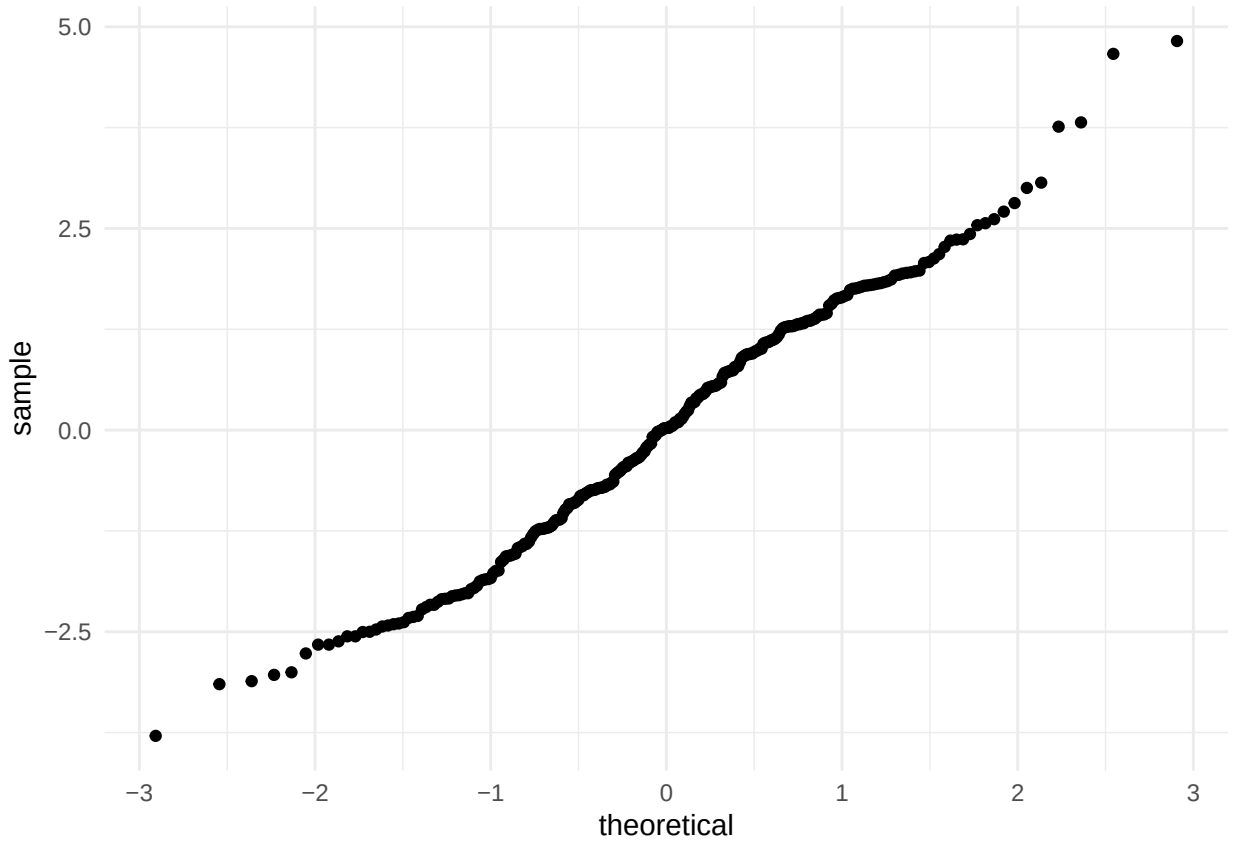
```
fit1 <- lm(LIFESAT_I ~ SOCIALWB_I, data=filter_data)
summary(fit1)
```

```
##
## Call:
## lm(formula = LIFESAT_I ~ SOCIALWB_I, data = filter_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7908 -1.2070  0.0253  1.2716  4.8236
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.37093    0.45536   0.815    0.416
## SOCIALWB_I   0.79653    0.09976   7.984 3.99e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.57 on 272 degrees of freedom
## Multiple R-squared:  0.1899, Adjusted R-squared:  0.1869
## F-statistic: 63.75 on 1 and 272 DF,  p-value: 3.994e-14
```

```
filter_data %>%
  add_residuals(fit1, "resid") %>%
  ggplot(aes(x=SOCIALWB_I)) +
  geom_point(aes(y=resid), alpha=0.2) +
  labs(x="Social Well being", y="Residuals") +
  theme_minimal()
```




```
filter_data %>%
  add_residuals(fit1, "resid") %>%
  ggplot(aes(sample=resid)) +
  geom_qq() +
  theme_minimal()
```



From the plot we can see it is not systematic. There isn't a very symmetric pattern, so we can consider no violations of model assumptions. Distribution is also approximately normal. So we can take Social well being as a predictor.

Seeing high error points.

```
filter_data %>%
  add_residuals(fit1, "resid") %>%
  filter(resid > 3 | resid < -3)
```

```
##    LIFESAT_I SOCIALWB_I NONAFFIRM_I NONDISCLOSURE_I HCTHREAT_I KESSLER6_I
## 1        1.2   5.800000    2.833333             5.0       4.00          5
## 2        2.0   6.000000    2.000000             3.4       3.25          7
## 3        7.0   3.533333    2.333333             2.0       5.00          0
## 4        1.0   4.600000    4.500000             3.6       4.75         20
## 5        7.0   2.466667    3.166667             1.4       1.00          0
## 6        2.2   6.066667    3.166667             4.4       3.50         15
## 7        6.2   3.466667    3.833333             4.0       4.75          3
## 8        7.0   3.600000    1.000000             1.0       1.00          0
## 9        6.4   3.800000    1.000000             3.6       1.00          5
## 10       7.0   2.266667    2.166667             3.2       5.00         21
## 11       1.4   5.200000    4.500000             4.8       5.00         19
##    EVERYDAY_I     resid
## 1    1.666667 -3.790789
## 2    1.333333 -3.150095
## 3    1.000000  3.814675
## 4    3.222222 -3.034955
## 5    1.000000  4.664305
## 6    3.666667 -3.003197
```

```
## 7    2.222222  3.067776
## 8    1.000000  3.761573
## 9    1.111111  3.002267
## 10   1.000000  4.823610
## 11   3.222222 -3.112872
```

Removing high error samples can improve the model fit and reduce the influence of outliers on the estimated regression coefficients. However it may not be the best way as we don't have a good understanding of underlying model assumptions. To be sure, checking if removing them does improve model performance.

```
new <- filter_data %>%
  add_residuals(fit1, "resid") %>%
  filter(resid <= 3 & resid >= -3)

new <- subset(new, select = -resid)

fit1 <- lm(LIFESAT_I ~ SOCIALWB_I, data=new)
summary(fit1)
```

```
##
## Call:
## lm(formula = LIFESAT_I ~ SOCIALWB_I, data = new)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.77774 -0.97531  0.02712  1.10673  3.04411
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.62032    0.42470  -1.461    0.145
## SOCIALWB_I   1.01214    0.09295  10.890   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.405 on 261 degrees of freedom
## Multiple R-squared:  0.3124, Adjusted R-squared:  0.3098
## F-statistic: 118.6 on 1 and 261 DF,  p-value: < 2.2e-16
```

```
sampled_data <- resample_partition(filter_data, p=c(train=0.6, valid=0.2, test=0.2))
```

```
fit1 <- lm(LIFESAT_I~SOCIALWB_I, data=sampled_data$train)
rmse(fit1, sampled_data$valid)
```

```
## [1] 1.606932
```

```
rmse(fit1, sampled_data$test)
```

```
## [1] 1.54662
```

```
sampled_data <- resample_partition(new, p=c(train=0.6, valid=0.2, test=0.2))
```

```
fit1 <- lm(LIFESAT_I~SOCIALWB_I, data=sampled_data$train)
rmse(fit1, sampled_data$valid)
```

```
## [1] 1.408347
```

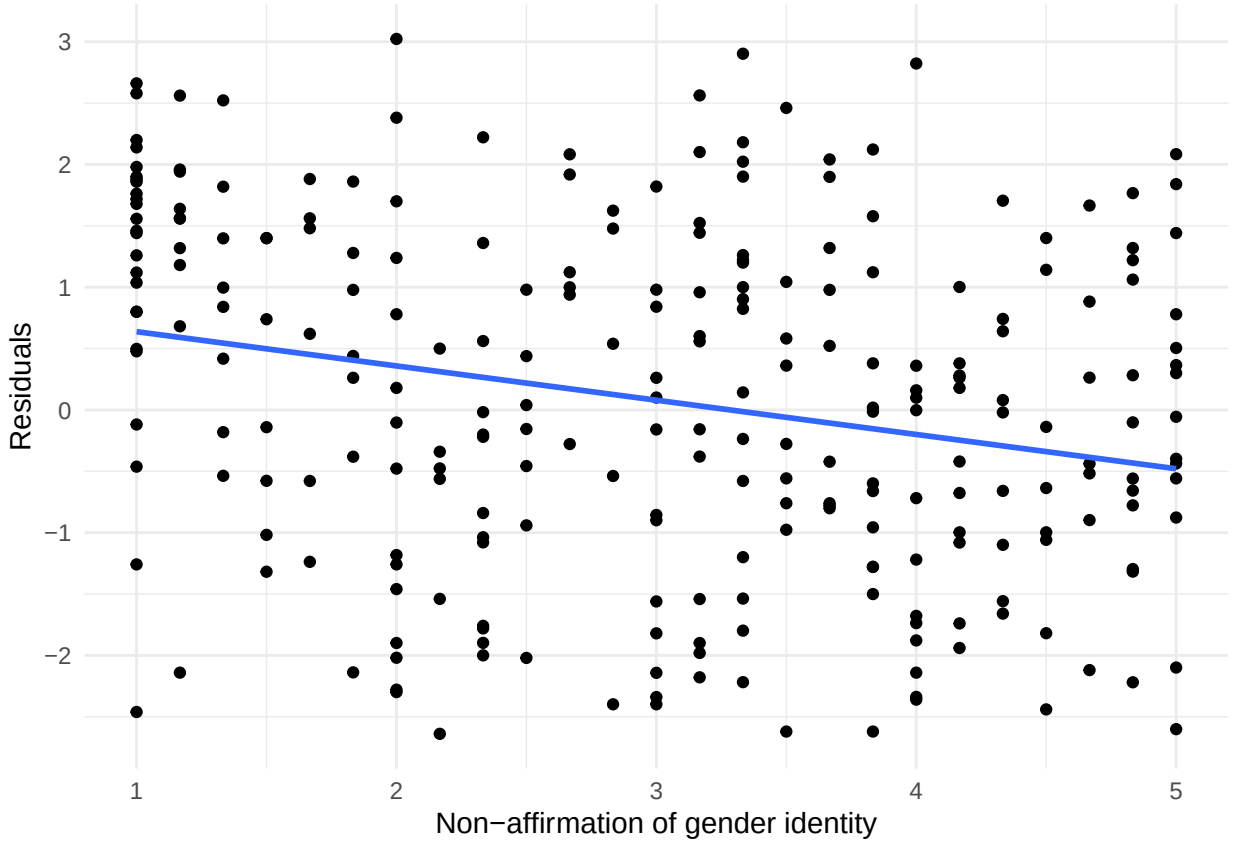
```
rmse(fit1, sampled_data$test)
```

```
## [1] 1.468145
```

Removing high error samples does improve the model fit. So using that data further.

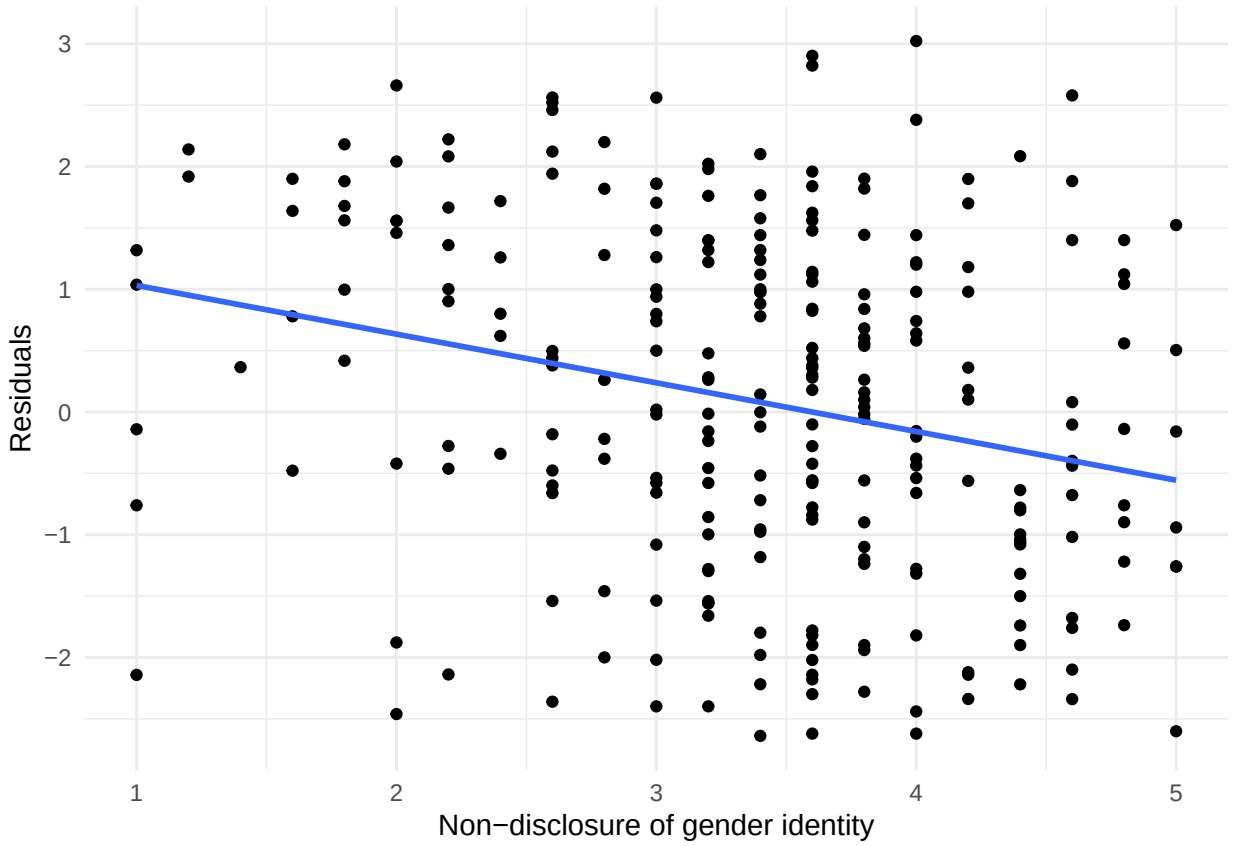
```
new %>%
  add_residuals(fit1, "resid") %>%
  ggplot(aes(x=NONAFFIRM_I, y=resid)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  labs(x="Non-affirmation of gender identity", y="Residuals") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



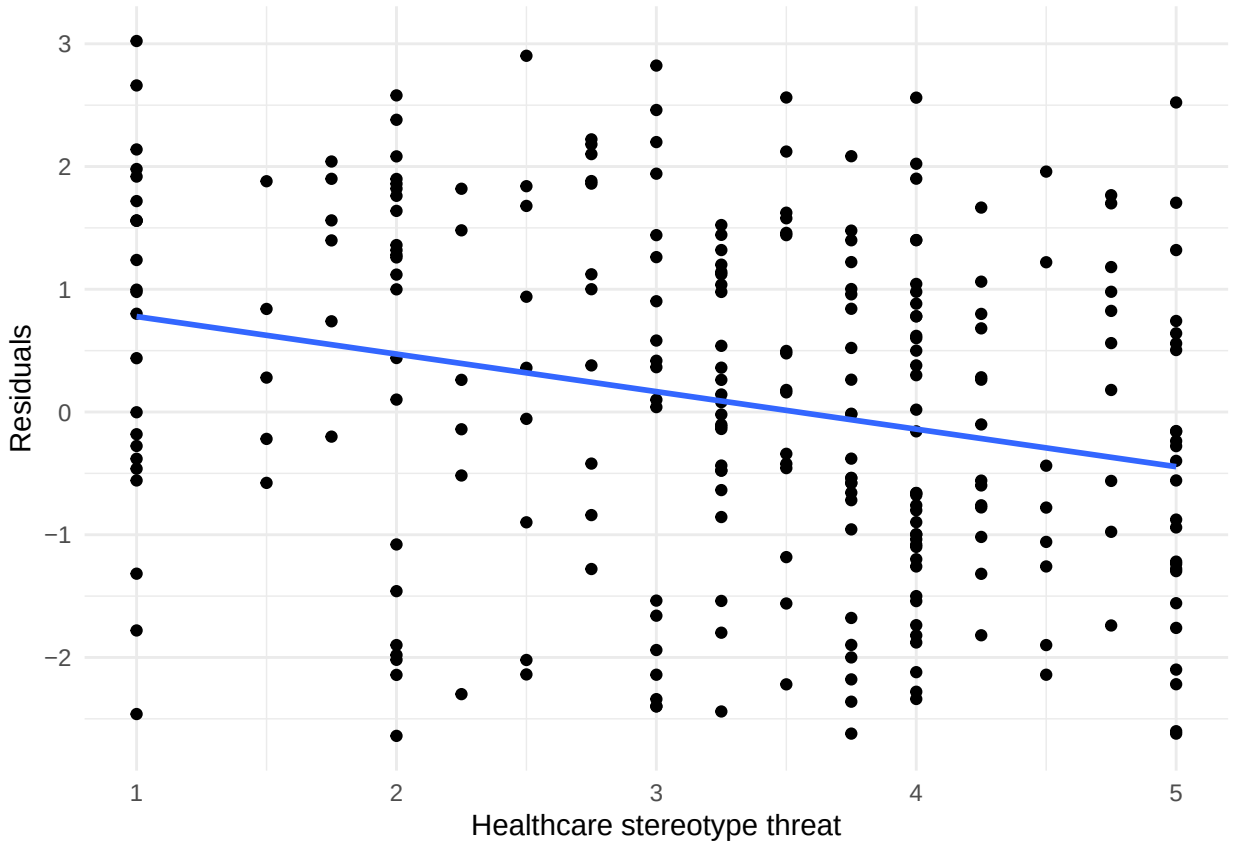

```
new %>%
  add_residuals(fit1, "resid") %>%
  ggplot(aes(x=NONDISCLOSURE_I, y=resid)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  labs(x="Non-disclosure of gender identity", y="Residuals") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



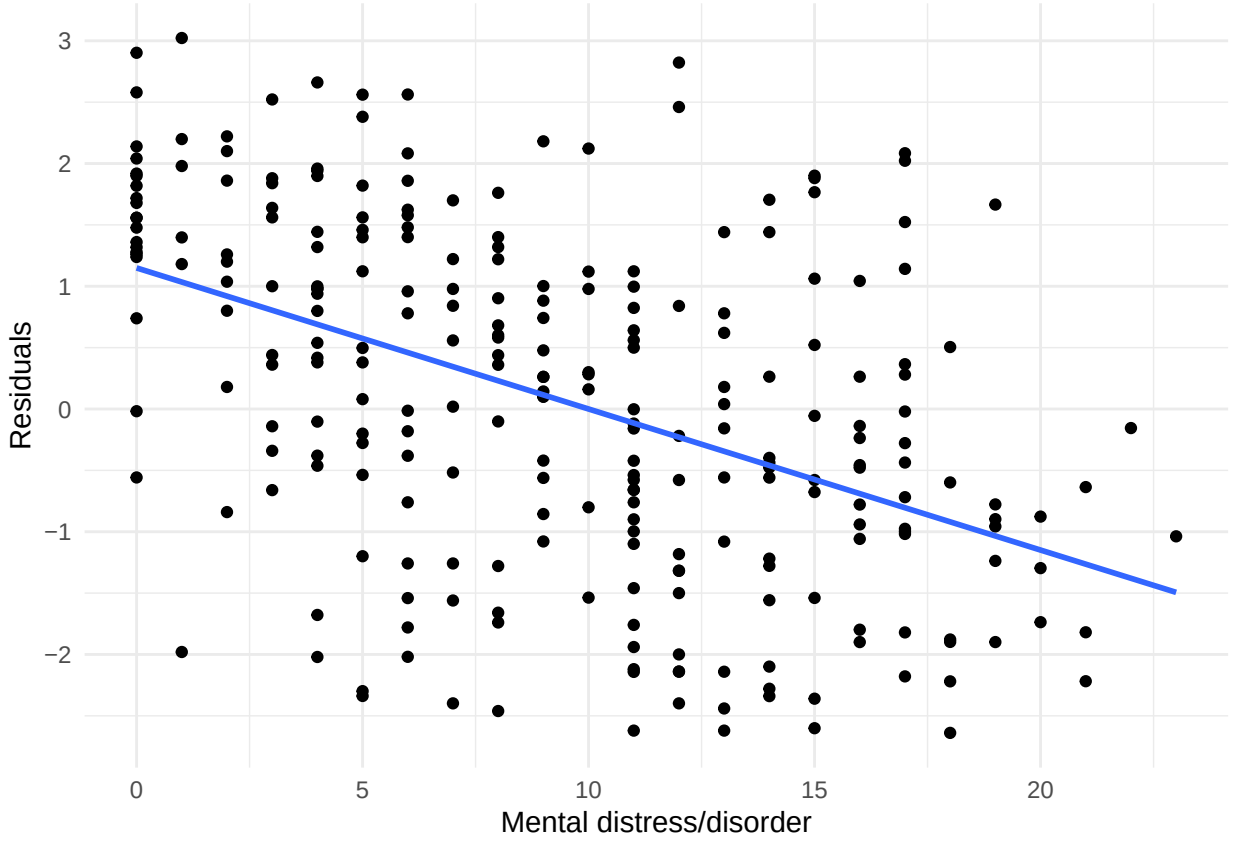

```
new %>%
  add_residuals(fit1, "resid") %>%
  ggplot(aes(x=HCTHREAT_I, y=resid)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  labs(x="Healthcare stereotype threat", y="Residuals") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



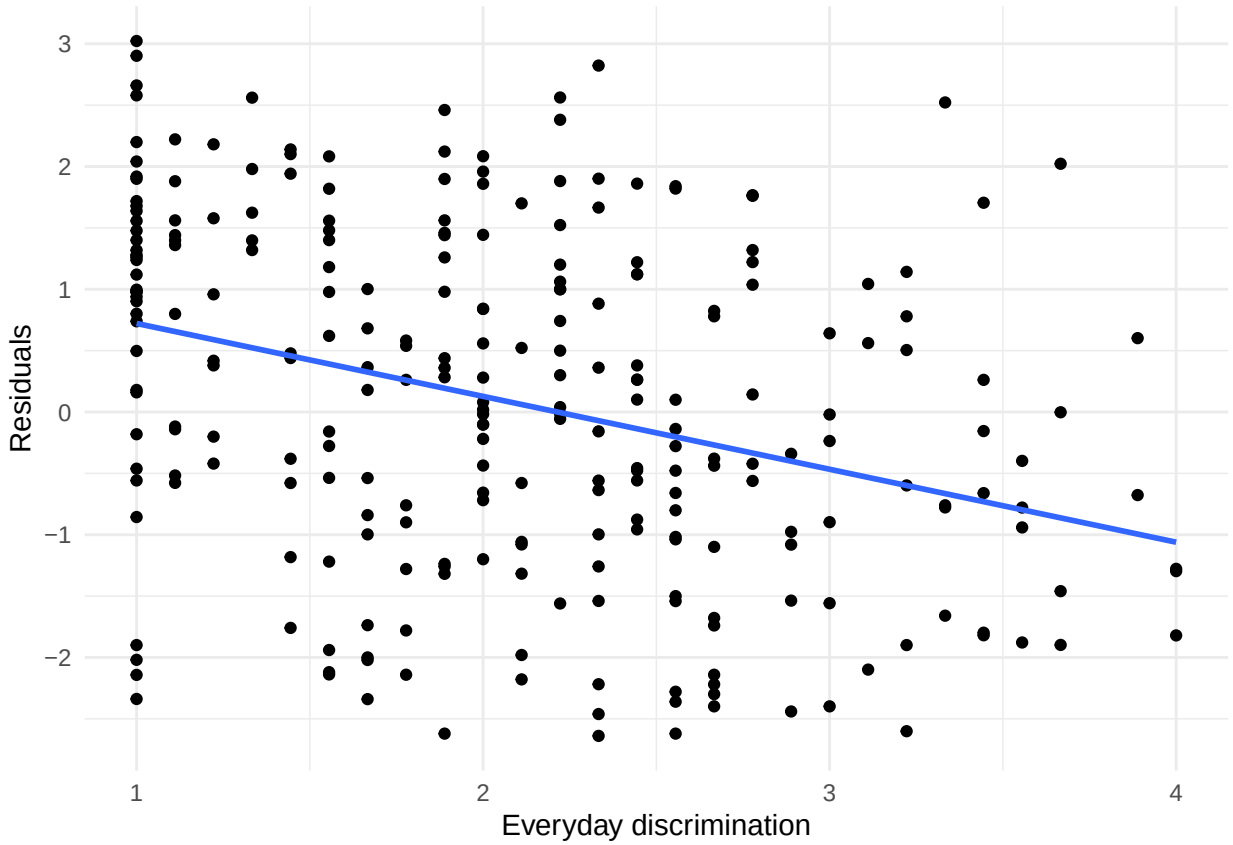

```
new %>%
  add_residuals(fit1, "resid") %>%
  ggplot(aes(x=KESSLER6_I, y=resid)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  labs(x="Mental distress/disorder", y="Residuals") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
new %>%
  add_residuals(fit1, "resid") %>%
  ggplot(aes(x=EVERYDAY_I, y=resid)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  labs(x="Everyday discrimination", y="Residuals") +
  theme_minimal()
```

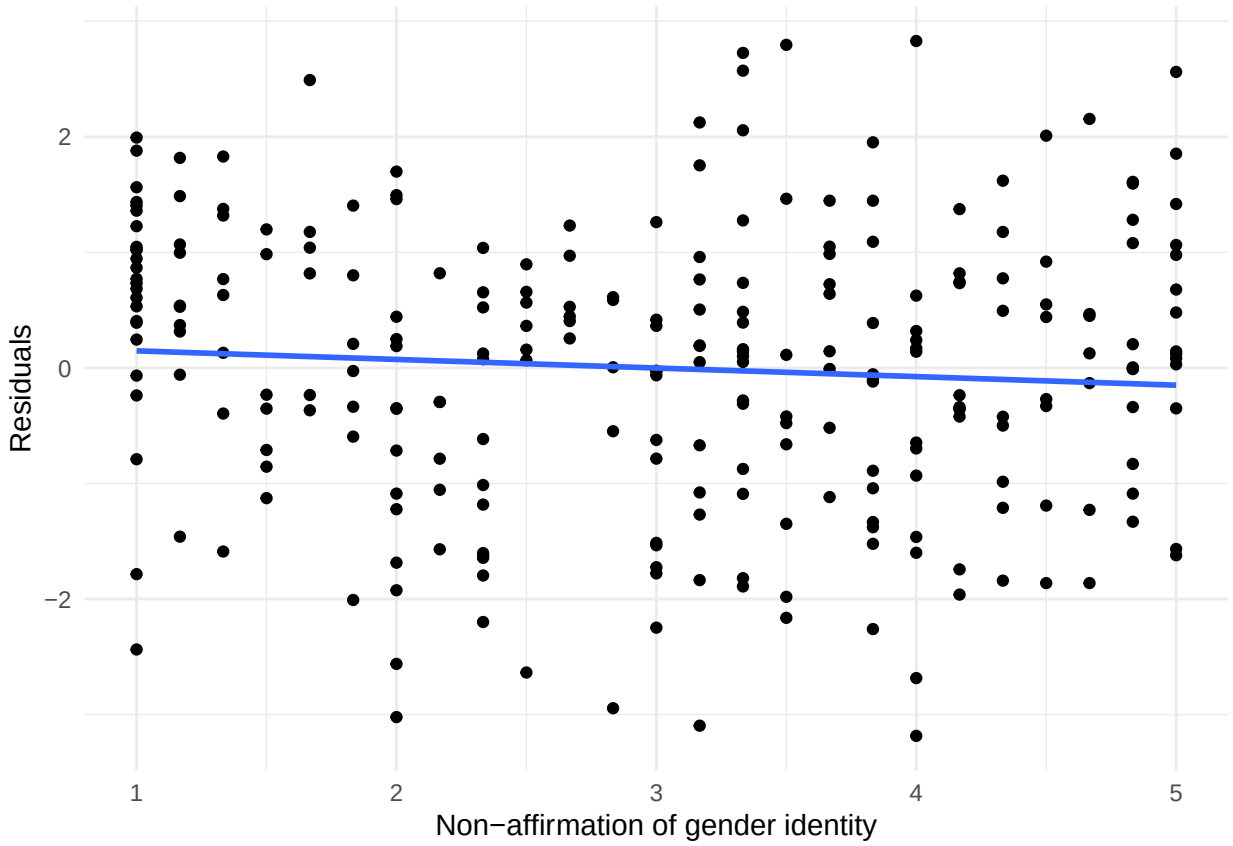
```
## `geom_smooth()` using formula = 'y ~ x'
```



From above residuals, mental distress (KESSLER6_I) seems to be systematic (has a pattern) so this indicates a violation of model assumption.

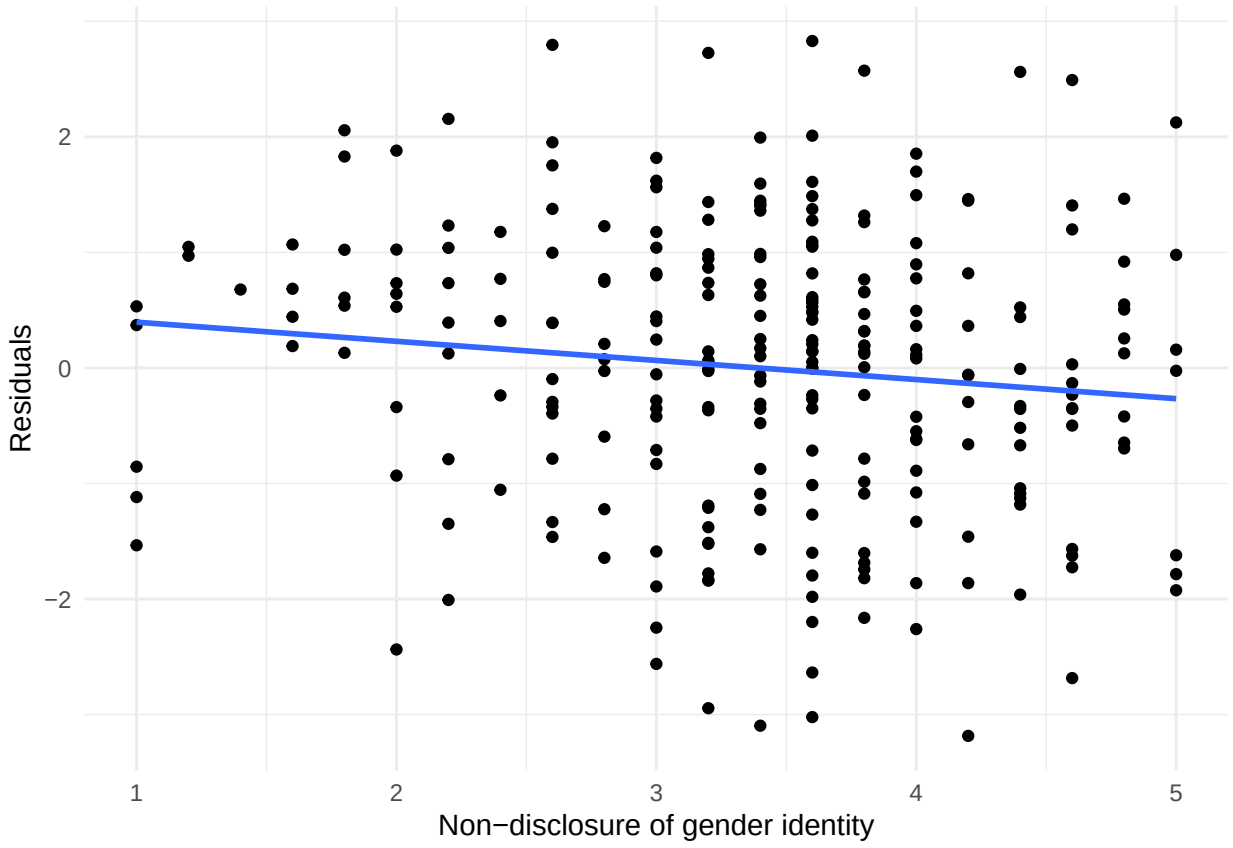
```
fit2 <- lm(LIFESAT_I~SOCIALWB_I + KESSLER6_I, data=new)

new %>%
  add_residuals(fit2, "resid") %>%
  ggplot(aes(x=NONAFFIRM_I, y=resid)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  labs(x="Non-affirmation of gender identity", y="Residuals") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



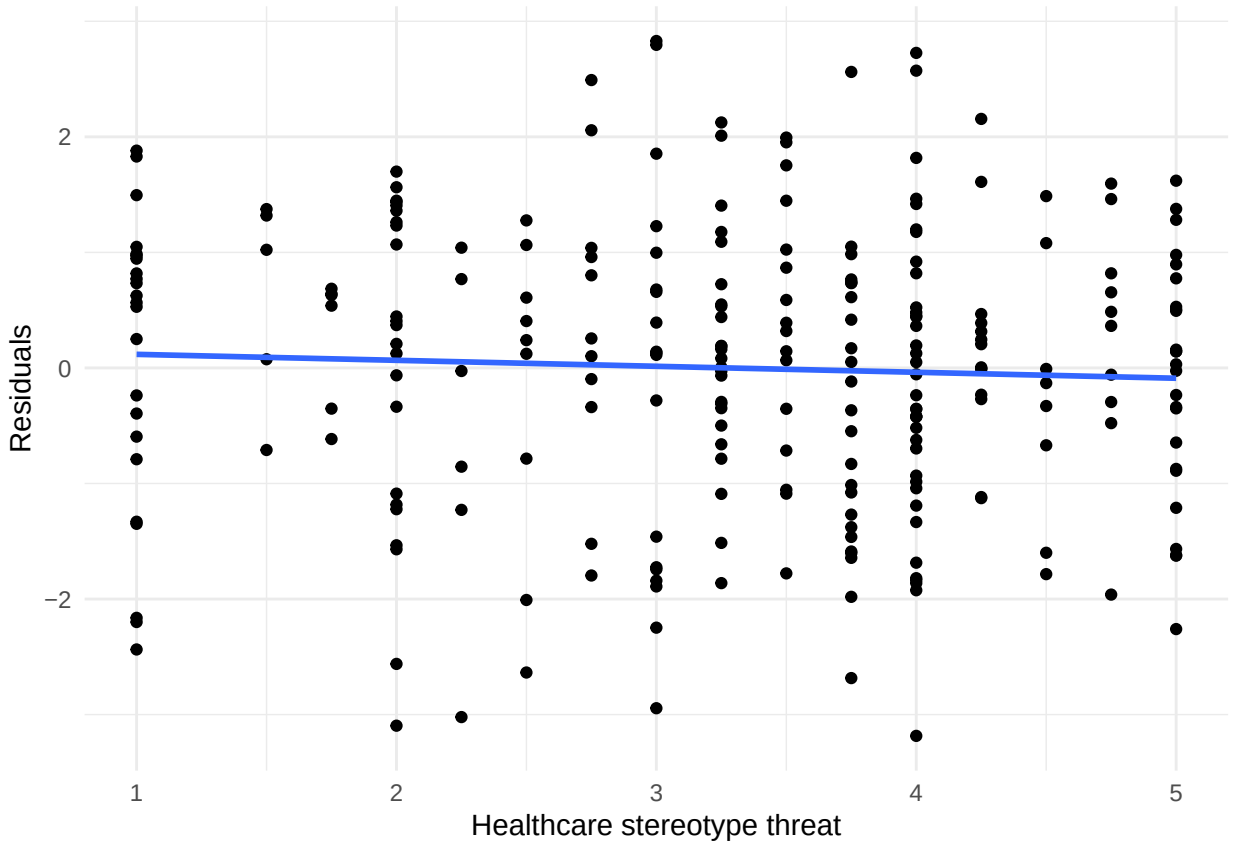

```
new %>%
  add_residuals(fit2, "resid") %>%
  ggplot(aes(x=NONDISCLOSURE_I, y=resid)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  labs(x="Non-disclosure of gender identity", y="Residuals") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



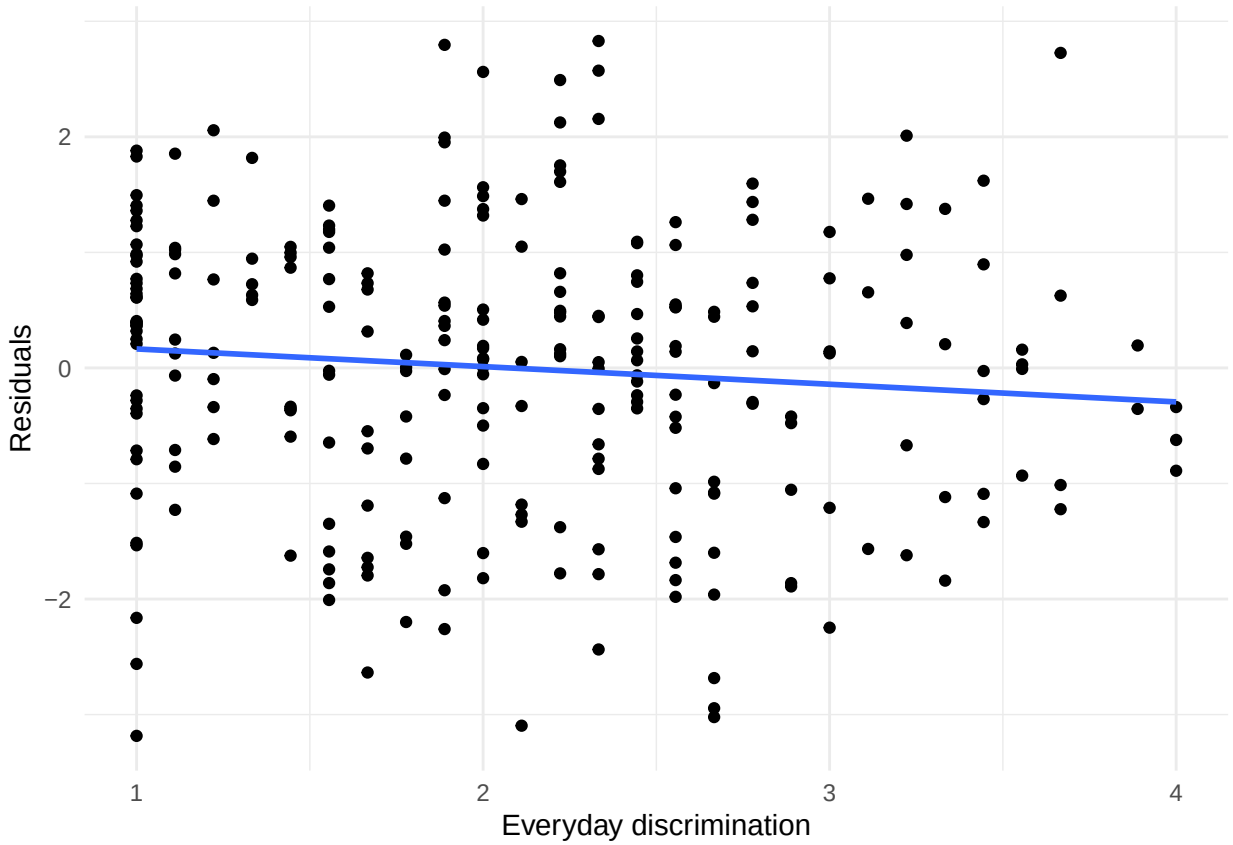

```
new %>%
  add_residuals(fit2, "resid") %>%
  ggplot(aes(x=HCTHREAT_I, y=resid)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  labs(x="Healthcare stereotype threat", y="Residuals") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
new %>%
  add_residuals(fit2, "resid") %>%
  ggplot(aes(x=EVERYDAY_I, y=resid)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  labs(x="Everyday discrimination", y="Residuals") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



Seeing the residuals, there is no violation of model assumption. So mental distress is added as a predictor.

Also if we check, model performs better (RMSE improves) when mental distress is added as a predictor.

```
step1 <- function(response, predictors, candidates, partition)
{
  rhs <- paste0(paste0(predictors, collapse="+"), "+", candidates)
  formulas <- lapply(paste0(response, "~", rhs), as.formula)
  rmses <- sapply(formulas, function(fm) rmse(lm(fm, data=partition$train),
                                    data=partition$valid))
  names(rmses) <- candidates
  attr(rmses, "best") <- rmses[which.min(rmses)]
  rmses
}
```

```
model <- NULL
response <- 'LIFESAT_I'
```

```
preds <- "SOCIALWB_I"
cands <- c('NONAFFIRM_I', 'NONDISCLOSURE_I',  'HCTHREAT_I', 'KESSLER6_I', 'EVERYDAY_I')
s1 <- step1(response, preds, cands, sampled_data)

model <- c(model, attr(s1, "best"))
model
```

```
## KESSLER6_I
##   1.252868
```

```
s1
```

```
##     NONAFFIRM_I NONDISCLOSURE_I      HCTHREAT_I      KESSLER6_I      EVERYDAY_I
##        1.318274        1.405481        1.318964        1.252868        1.357773
## attr(,"best")
## KESSLER6_I
##   1.252868
```

```
fit2 <- lm(LIFESAT_I~SOCIALWB_I + KESSLER6_I, data=sampled_data$train)

rmse(fit2, sampled_data$valid)
```

```
## [1] 1.252868
```

```
rmse(fit2, sampled_data$test)
```

```
## [1] 1.270731
```