IDMP Assignment 4

Mansi Pravin Thanki

2023-03-14

```
# Install stringr R package
install.packages("stringr", repos = "http://cran.us.r-project.org")
```

```
##
## The downloaded binary packages are in
##  /var/folders/r9/2cgj8871421bvklfhk05xfzc0000gn/T//Rtmp8w1Zcg/downloaded_packages
```

```
install.packages("tidyverse", repos = "http://cran.us.r-project.org")
```

```
##
## The downloaded binary packages are in
##  /var/folders/r9/2cgj8871421bvklfhk05xfzc0000gn/T//Rtmp8w1Zcg/downloaded_packages
```

```
# Load required packages
library("stringr") # Load stringr R package
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.0     v purrr     1.0.1
## v forcats   1.0.0     v readr     2.1.4
## v ggplot2   3.4.1     v tibble    3.1.8
## v lubridate 1.9.2     v tidyr     1.3.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(ggplot2)
library(dplyr)
library(modelr)
```

#Loading dataset

```
load('/Users/mansipravinthanki/Downloads/TransPopData/37938-0001-Data.rda')
```

```
trans_data <- da37938.0001
as.tibble(trans_data)
```

```
## Warning: ‘as.tibble()‘ was deprecated in tibble 2.0.0.
## i Please use ‘as_tibble()‘ instead.
## i The signature and semantics have changed, see ‘?as_tibble‘.
```

```
## # A tibble: 274 x 566
##      STUDYID WEIGHT SURVEY~1 GRESP~2 GCENREG   AGE RACE  RACE_~3 RACE_~4 SEXUA~5
##        <dbl>  <dbl> <fct>    <fct>   <fct>   <dbl> <fct> <fct>   <fct>   <fct>
##  1 151768927  0.986 (0) Tra~ 26-APR~ (1) No~    65 (6) ~ (1) Wh~ (1) Wh~ (1) St~
##  2 152357242  0.380 (0) Tra~ 07-APR~ (3) So~    38 (6) ~ (1) Wh~ (1) Wh~ (1) St~
##  3 152444055  0.705 (0) Tra~ 01-MAY~ (3) So~    25 (6) ~ (1) Wh~ (1) Wh~ (5) Qu~
##  4 152525272  1.60  (0) Tra~ 20-APR~ (4) We~    18 (6) ~ (1) Wh~ (1) Wh~ (3) Gay
##  5 152894493  1.87  (0) Tra~ 05-MAY~ (2) Mi~    30 (8) ~ (3) La~ (3) La~ (1) St~
##  6 152925625  0.955 (0) Tra~ 10-JUN~ (1) No~    64 (8) ~ (6) Mu~ (4) Mu~ (1) St~
##  7 153003265  0.236 (0) Tra~ 29-APR~ (3) So~    61 (6) ~ (1) Wh~ (1) Wh~ (3) Gay
##  8 153036828  0.444 (0) Tra~ 18-JUL~ (2) Mi~    57 (8) ~ (6) Mu~ (4) Mu~ (7) Ot~
##  9 153162357  1.09  (0) Tra~ 31-MAY~ (3) So~    54 (2) ~ (2) Bl~ (2) Bl~ (6) Sa~
## 10 153318257  0.939 (0) Tra~ 06-MAY~ (4) We~    21 (6) ~ (1) Wh~ (1) Wh~ (4) Bi~
## # ... with 264 more rows, 556 more variables: SEXMINID <fct>, SEX <fct>,
## #   GENDER <fct>, HINC <fct>, HINC_I <fct>, PINC <fct>, PINC_I <fct>,
## #   GEDUC1 <fct>, GEDUC2 <fct>, GURBAN <fct>, GURBAN_I <fct>, POVERTY <fct>,
## #   POVERTY_I <fct>, POVERTYCAT <fct>, POVERTYCAT_I <fct>, GANN_INC <fct>,
## #   GANN_INC2 <fct>, GD74 <fct>, GD75 <fct>, GD76 <fct>, GEDUCATION <fct>,
## #   GEMPLOYMENT2010 <fct>, GMSANAME <fct>, GRACE <fct>, GZIPCODE <dbl>,
## #   GZIPSTATE <dbl>, GD7 <fct>, GRUCA <dbl>, GRUCA_I <dbl>, ...
```

Problem 1

We would like to compare the weighted and unweighted distributions of trans people of different races and ethnicities. Visualize bar plots showing (1) the unweighted proportions of trans people of each race/ethnicity and (2) the weighted proportions of trans people of each race/ethnicity. What races are over- or underrepresented in the survey sample compared to the population?

Cleaning the data

```
# cleaning the data to remove bracketed numbers
trans_data$RACE <- substring(trans_data$RACE, 4)
```

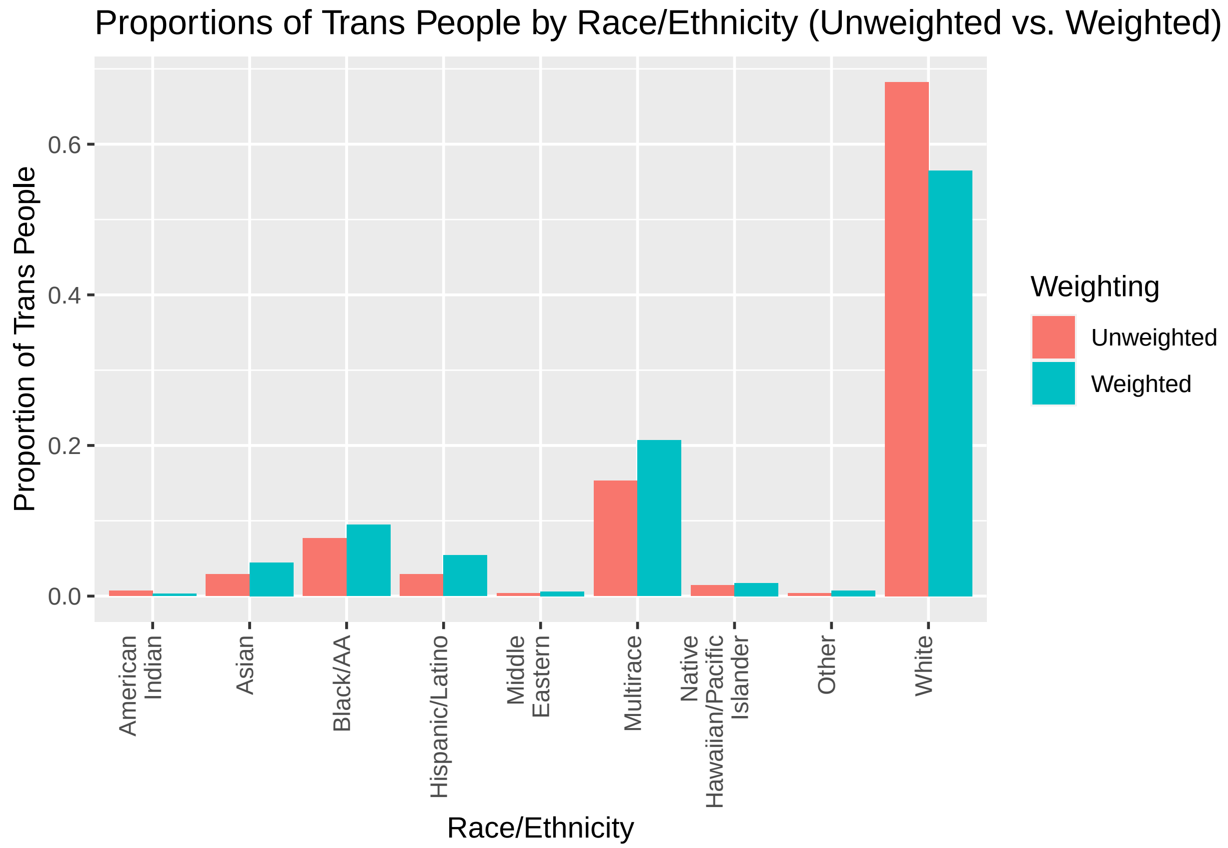
Calculate weighted and unweighted proportions of trans people of each race/ethnicity

```
# Load required packages
library(ggplot2)
library(dplyr)

# Subset the data frame to include WEIGHT and RACE
transpop <- trans_data %>% select(WEIGHT, RACE)

# Calculate unweighted proportions of trans people of each race/ethnicity
unweighted_prop <- transpop %>% count(RACE) %>%
  mutate(prop = n / sum(n))
```

```r
# Calculate weighted proportions of trans people of each race/ethnicity
weighted_prop <- transpop %>%
  group_by(RACE) %>%
  summarize(total_weight = sum(WEIGHT)) %>%
  mutate(prop = total_weight / sum(total_weight))

# Combine unweighted and weighted proportions into a single data frame
prop_df <- bind_rows(
  data.frame(type = "Unweighted", RACE = unweighted_prop$RACE, prop = unweighted_prop$prop),
  data.frame(type = "Weighted", RACE = weighted_prop$RACE, prop = weighted_prop$prop)
)
```

Visualization for Proportions of Trans People by Race/Ethnicity (Unweighted vs. Weighted)

```r
# Create bar plots
ggplot(prop_df, aes(x = RACE, y = prop, fill = type)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(x = "Race/Ethnicity", y = "Proportion of Trans People", fill = "Weighting") +
  ggtitle("Proportions of Trans People by Race/Ethnicity (Unweighted vs. Weighted)")+
  theme(axis.text.x = element_text(angle = 90, vjust = 0.2, hjust=1))+
  scale_x_discrete(labels = function(x) str_wrap(x, width = 5))
```



#Comments and Observations:

1. The **White Race and American Indian races** are over-represented in the survey sample compared to the population. This is because the unweighted proportion is higher than the weighted proportion.
2. The **Asian, Black/AA, Hispanic/Latino, Middle Eastern, Multirace, Native Hawaiian/Pacific Islander and Other** races are under represented in the survey sample compared to the population. This is because the weighted proportion is higher than the unweighted proportion.
3. The above 2 points suggests the presence of sampling bias and unit non-response.
4. The **White race and Multirace** have significantly higher proportion in both weighted and non-weighted proportions compared to other races in the population

5.The **American Indian, Middle Eastern and Other** races have significantly low poroprtions in both weighted and non-weighted proportions compared to other races in the population

Summing weighted proportions to check if it adds 1

```
sum(weighted_prop$prop)
```

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

Summing unweighted proportions to check if it adds 1

```
sum(unweighted_prop$prop)
```

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

Problem 2

We would like to compare the weighted and unweighted distributions of trans people with different sexual orientations. Visualize bar plots showing (1) the unweighted proportions of trans people of each sexual orientation and (2) the weighted proportions of trans people of each sexual orientation. What sexual identities are over- or under-represented in the survey sample compared to the population?

Cleaning the data

```
# cleaning the data to remove bracketed numbers
trans_data$SEXUALID <- substring(trans_data$SEXUALID, 4)
```

Calculating weighted and unweighted counts and proportion

```r
library(dplyr)

# Calculate unweighted counts and proportions of trans people of each sexual orientation
unweighted_counts <- trans_data %>%
  group_by(SEXUALID) %>%
  summarize(count = n())

unweighted_props <- unweighted_counts %>%
  mutate(prop = count/sum(count))

# Calculate weighted counts and proportions of trans people of each sexual orientation
weighted_counts <- trans_data %>%
  group_by(SEXUALID)%>%
  summarize(count = sum(WEIGHT))

weighted_props <- weighted_counts %>%
  mutate(prop = count/sum(count))
```

Combine unweighted and weighted proportions into a single data frame

```r
# Combine unweighted and weighted proportions into a single data frame
proportion_df <- bind_rows(
  data.frame(type = "Unweighted",
             SEXUALID = unweighted_props$SEXUALID, prop = unweighted_props$prop),
  data.frame(type = "Weighted",
             SEXUALID = weighted_props$SEXUALID, prop = weighted_props$prop)
)
```

Summing weighted proportions to check if it adds 1

```r
sum(weighted_props$prop)
```

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

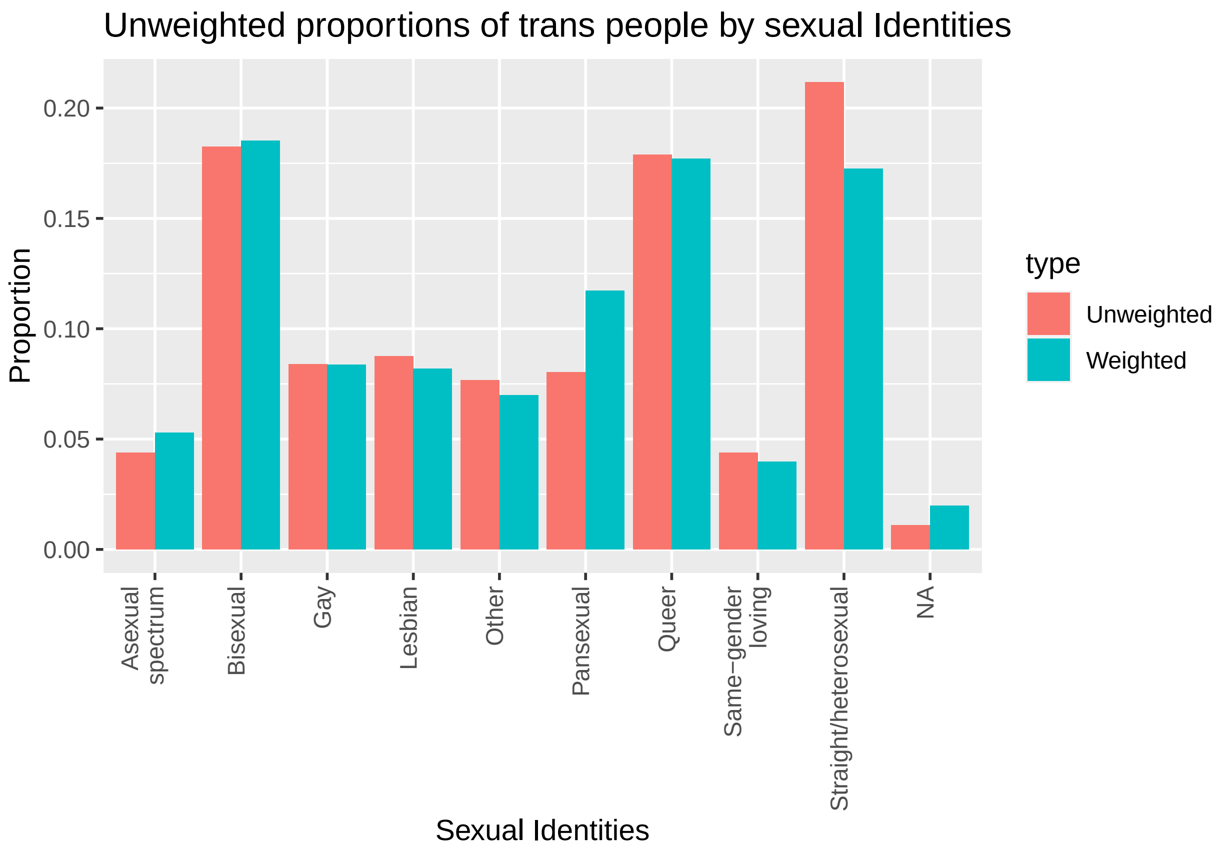
Summing unweighted proportions to check if it adds 1

```r
sum(unweighted_props$prop)
```

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

I have included the NA category in the plot since sum of proportion is adding to 1

```r
library(ggplot2)

# Bar plot
ggplot(proportion_df, aes(x = SEXUALID, y = prop, fill = type)) +
  geom_bar(stat = "identity", position = "dodge") +
  xlab("Sexual Identities") +
  ylab("Proportion") +
  ggtitle("Unweighted proportions of trans people by sexual Identities")+
  theme(axis.text.x = element_text(angle = 90, vjust = 0.2, hjust=1))+
  scale_x_discrete(labels = function(x) str_wrap(x, width = 5))
```



Comments and Observations:

1. The sexual identities: **Straight/Heterosexual, Same-gender loving, Queer, Lesbian and Other** are over-represented in the survey sample compared to the population. This is because their unweighted proportion is higher than the weighted proportion
2. The sexual identities: **Asexual Spectrum, Bisexual and Pansexual** are under-respresented in the survey sample compared to the population. This is because their weighted proportion is higher than the unweighted proportion
3. The sexual identity: **Gay** is correctly represented in the survey sample compared to the population because its weighted and unweighted proportions are nearly same. This means there is no sampling bias or unit non-response for sexual identity Gay.
4. **Straight/Heterosexual, Bisexual and Queer** sexual identities have significantly higher proportion than other sexual identites in the population
5. **Asexual Spectrum and Same-gender loving** sexual identities have significantly lower proportion than other sexual identites in the population

Problem 3

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. Focus your analysis on the following numeric variables described on pages 26-35 of the User Guide:

- Satisfaction with life
- Social well-being
- Non-affirmation of gender identity
- Non-disclosure of gender identity
- Healthcare stereotype threat
- Mental distress/disorder
- Everyday discrimination

Using the imputed versions of these variables, visualize life satisfaction versus the six candidate predictors, transforming variables as necessary, and describe their relationships.

Selecting the variables of interest

```
# Select variables of interest
TransPopdata <- trans_data %>%
  select(STUDYID, LIFESAT_I, SOCIALWB_I, NONAFFIRM_I,
         NONDISCLOSURE_I, HCTHREAT_I, KESSLER6_I,
         EVERYDAY_I)
```

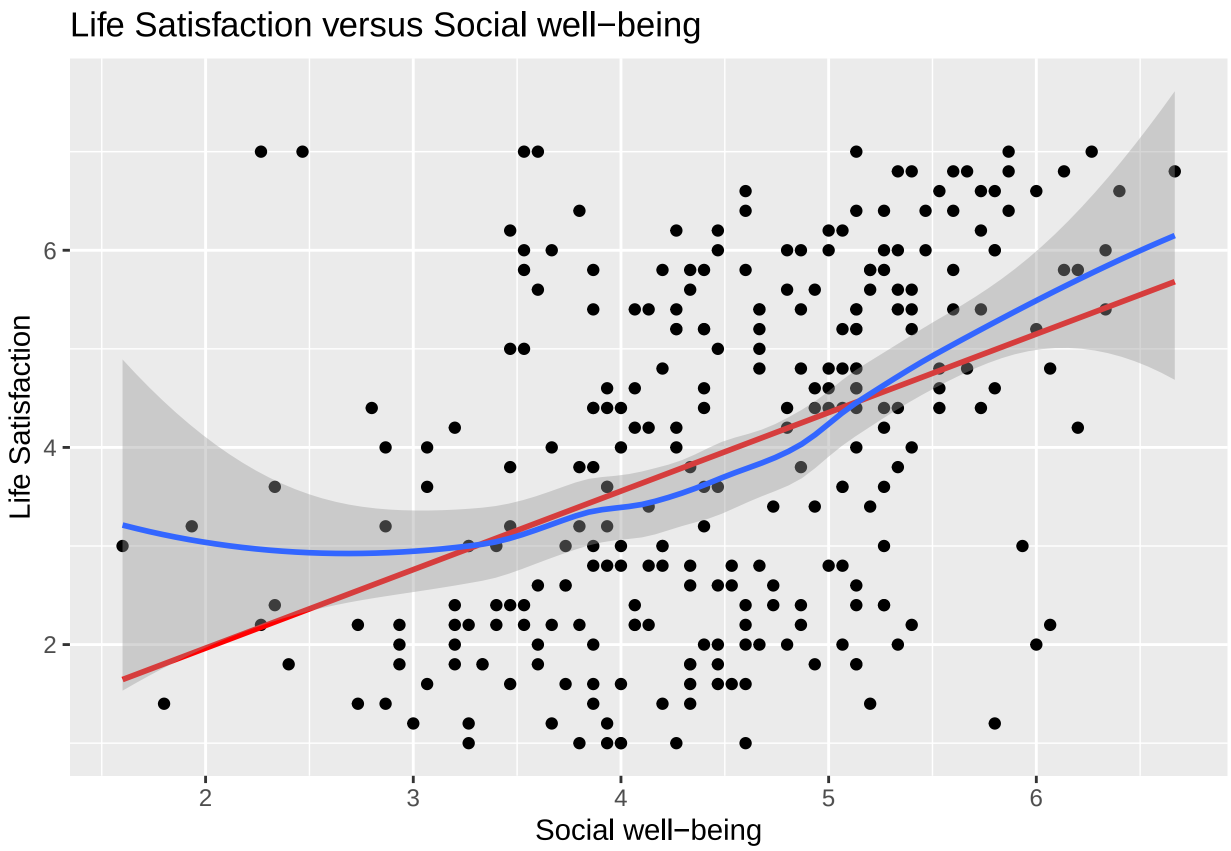
removing NA values from dataframe

```
TransPopdata <- na.omit(TransPopdata)
as.tibble(TransPopdata)
```

```
## # A tibble: 274 x 8
##      STUDYID LIFESAT_I SOCIALWB_I NONAFFIRM_I NONDISCL~1 HCTHR~2 KESSL~3 EVERY~4
##        <dbl>     <dbl>      <dbl>       <dbl>      <dbl>   <dbl>   <dbl>   <dbl>
##  1 151768927      2.20       4.87        3.17       3.40    2           1    2.11
##  2 152357242      6.20       4.27        1.17       3       4           5    1.33
##  3 152444055      4.20       4.27        2.33       3.80    4.75       11    3.11
##  4 152525272      1.80       3.20        5          3.60    5          20    2.44
##  5 152894493      5.40       4.27        1          3.20    2           8    2.78
##  6 152925625      6.40       5.60        1.17       2       1           0    1.56
##  7 153003265      6.80       5.40        1          1.20    1           0    1.44
##  8 153036828      4          3.67        3.33       2.20    3           8    1
##  9 153162357      2.40       4.60        2.17       2.60    3.25       15    2.33
## 10 153318257      2.20       5.40        1          2       1           8    2.33
## # ... with 264 more rows, and abbreviated variable names 1: NONDISCLOSURE_I,
## #   2: HCTHREAT_I, 3: KESSLER6_I, 4: EVERYDAY_I
```

```
ggplot(data = TransPopdata, aes(x = SOCIALWB_I, y = LIFESAT_I)) +
  geom_point() +
  stat_smooth(method=lm, color="red", se=FALSE)+
  geom_smooth() +
  ylab("Life Satisfaction") +
  xlab("Social well-being")+
  ggtitle("Life Satisfaction versus Social well-being ")
```

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

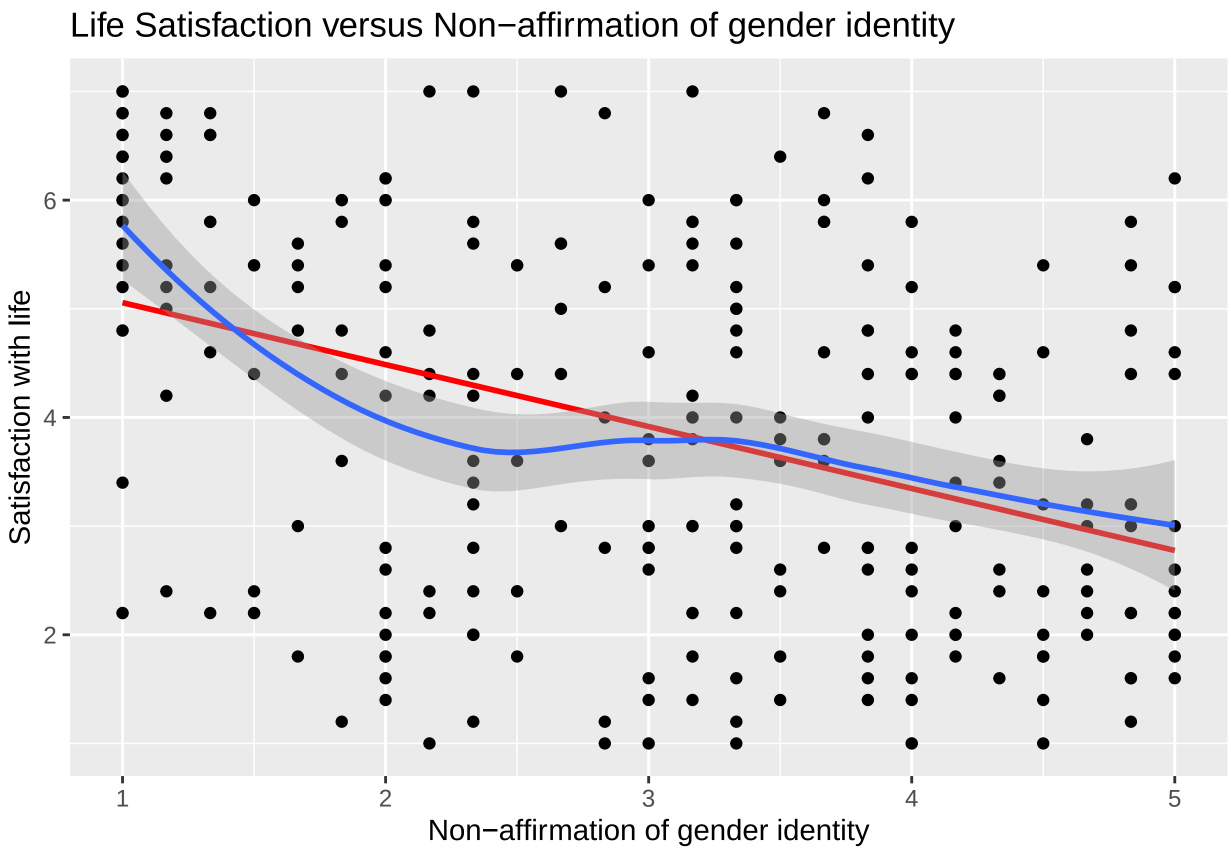


Comments on Visualization:

1. A strong **positive relationship** can be observed between social well-being and life satisfaction
2. The positive slope indicates that an increase in social well-being correlates to an increase in life satisfaction.
3. It is a directly positive linearly increasing relationship between well-being and life satisfaction

```
ggplot(data = TransPopdata, aes(x = NONAFFIRM_I, y = LIFESAT_I)) +
  geom_point() +
  stat_smooth(method=lm, color="red", se=FALSE)+
  geom_smooth() +
  ylab("Satisfaction with life") +
  xlab("Non-affirmation of gender identity")+
  ggtitle("Life Satisfaction versus Non-affirmation of gender identity")
```

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

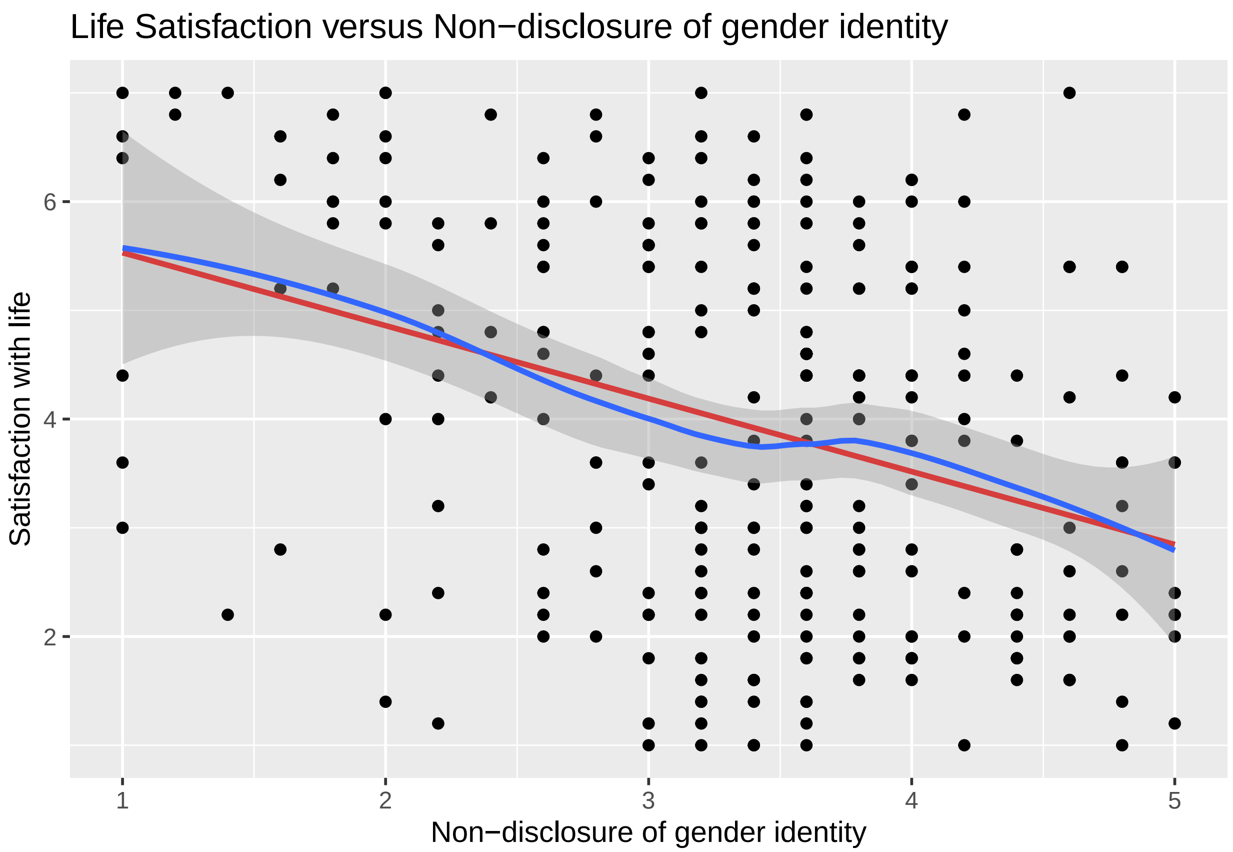


Comments on Visualization:

1. One can observe that there exists a **negative relationship** between non-affirmation of gender identity and life satisfaction
2. The negative slope indicates that increasing non-affirmation will lead to a decrease in life satisfaction.

```
ggplot(data = TransPopdata, aes(x = NONDISCLOSURE_I, y = LIFESAT_I)) +
  geom_point() +
  stat_smooth(method=lm, color="red", se=FALSE)+
  geom_smooth() +
  ylab("Satisfaction with life") +
  xlab("Non-disclosure of gender identity")+
  ggtitle("Life Satisfaction versus Non-disclosure of gender identity")
```

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

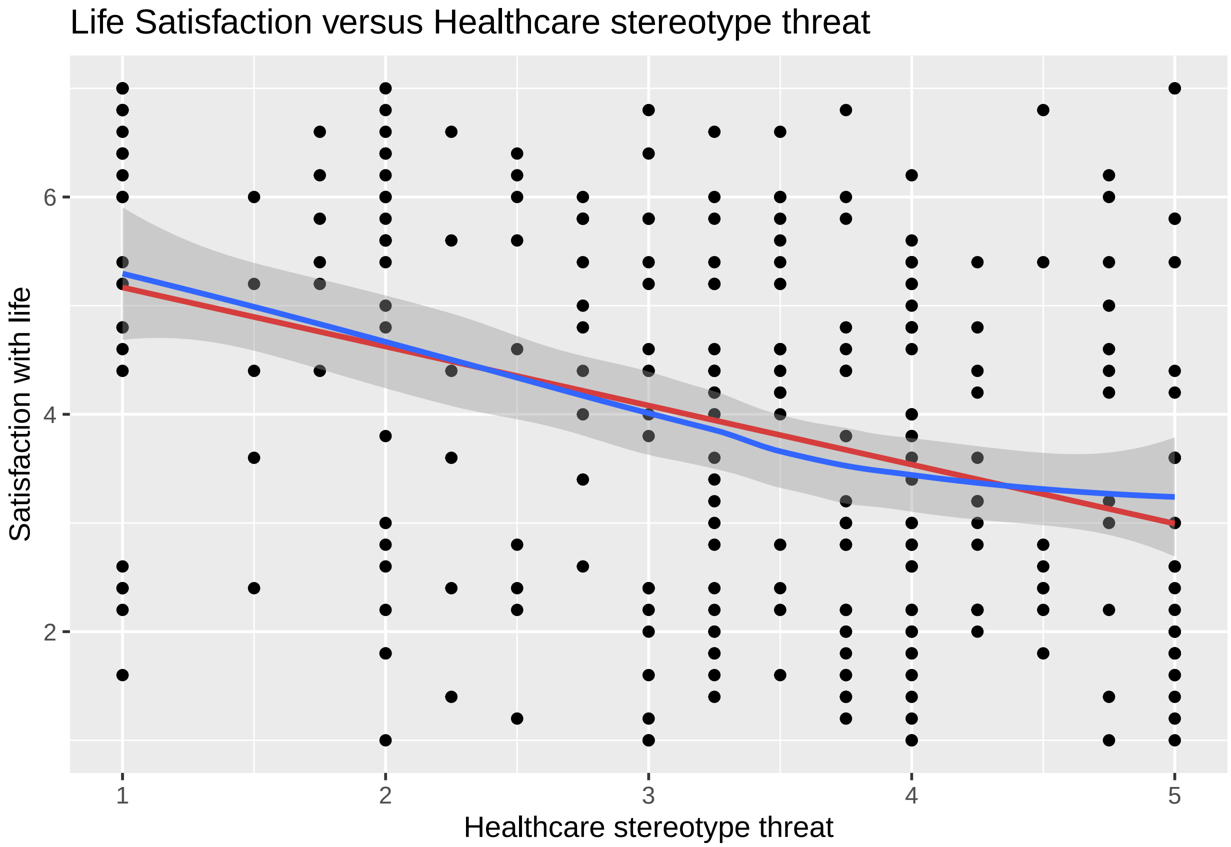


Comments on Visualization:

1. One can observe that there exists a **strong negative relationship** between Non-disclosure of gender identity and life satisfaction
2. The negative slope indicates that increasing Non-disclosure of gender identity will lead to a decrease in life satisfaction.

```
ggplot(data = TransPopdata, aes(x = HCTHREAT_I, y = LIFESAT_I)) +
  geom_point() +
  stat_smooth(method=lm, color="red", se=FALSE)+
  geom_smooth() +
  ylab("Satisfaction with life") +
  xlab("Healthcare stereotype threat")+
  ggtitle("Life Satisfaction versus Healthcare stereotype threat")
```

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

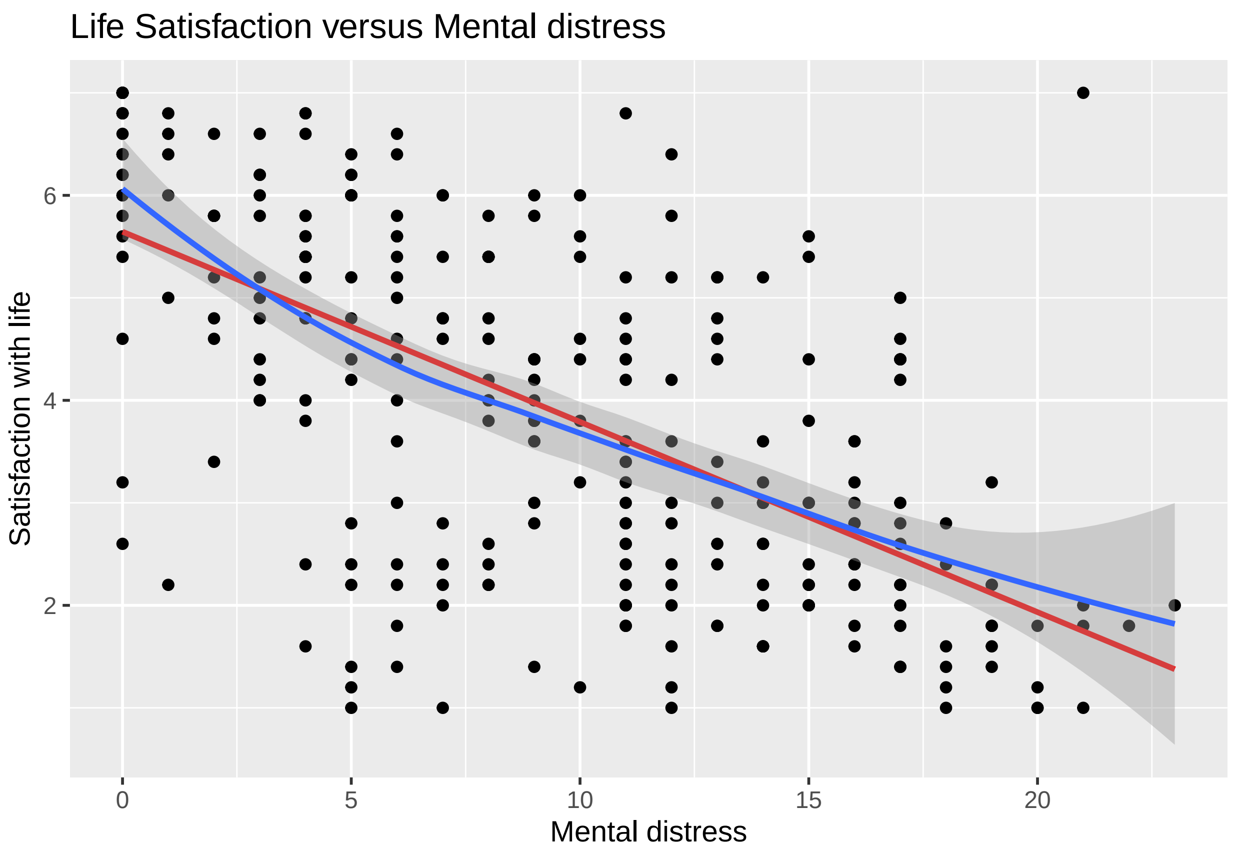


Comments on Visualization:

1. One can observe that there exists a **strong negative relationship** between Healthcare stereotype threat and life satisfaction
2. The negative slope indicates that increasing Healthcare stereotype threat will lead to a decrease in life satisfaction.

```
ggplot(data = TransPopdata, aes(x = KESSLER6_I, y = LIFESAT_I)) +
  geom_point() +
  stat_smooth(method=lm, color="red", se=FALSE)+
  geom_smooth() +
  ylab("Satisfaction with life") +
  xlab("Mental distress")+
  ggtitle("Life Satisfaction versus Mental distress")
```

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

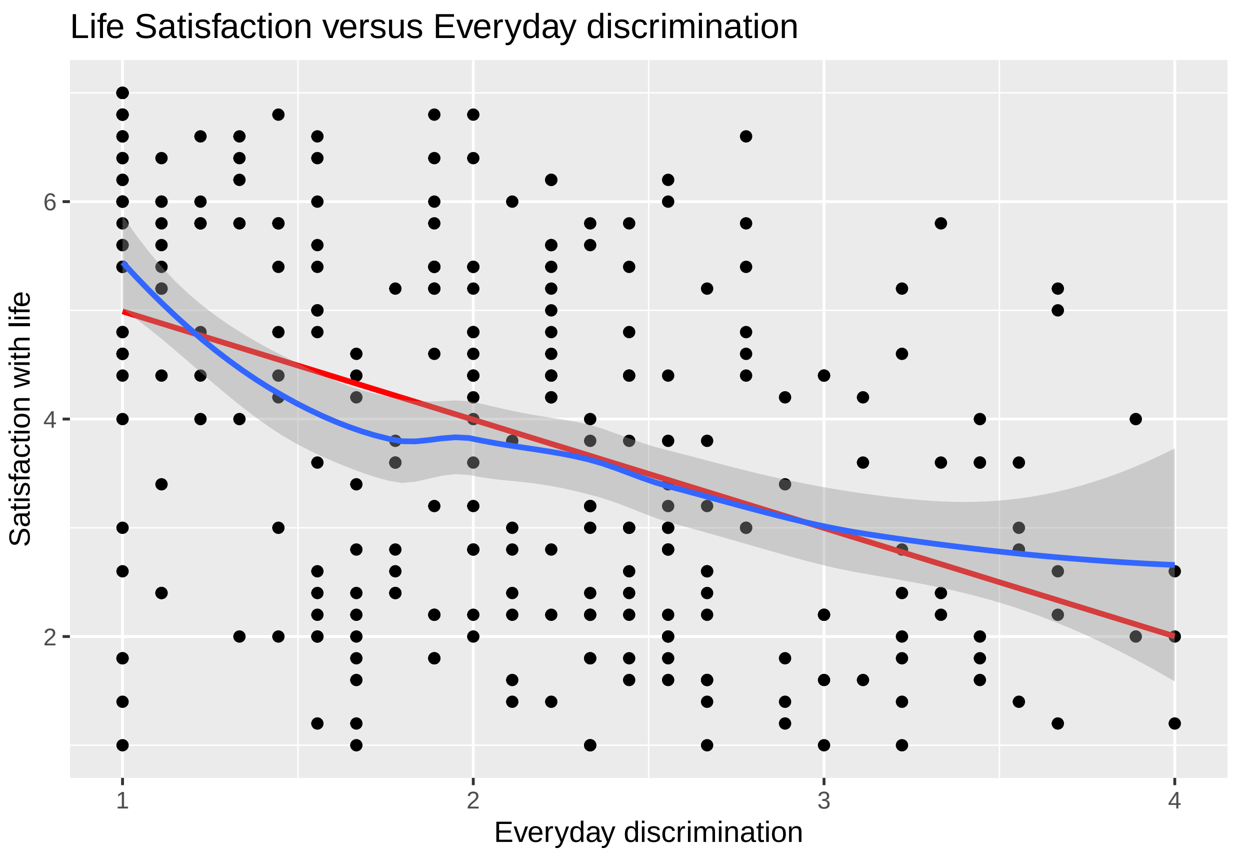


Comments on Visualization:

1. One can observe that there exists a **strong negative relationship** between Mental distress and life satisfaction
2. The negative slope indicates that increasing Mental distress will lead to a decrease in life satisfaction.

```
ggplot(data = TransPopdata, aes(x = EVERYDAY_I, y = LIFESAT_I)) +
  geom_point() +
  stat_smooth(method=lm, color="red", se=FALSE)+
  geom_smooth() +
  ylab("Satisfaction with life") +
  xlab("Everyday discrimination")+
  ggtitle("Life Satisfaction versus Everyday discrimination")
```

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



Comments on Visualization:

1. One can observe that there exists a **negative relationship** between Everyday discrimination and life satisfaction
2. The negative slope indicates that increasing Everyday discrimination will lead to a decrease in life satisfaction.

#Problem 4

Build a linear regression model for life satisfaction using a single predictor, justifying your choice based only on the visualizations from Problem 3. Then use residual plots to perform model diagnostics. Comment on any outliers or violations of model assumptions you notice in the residual plots. If necessary, fix the issue(s), re-fit the model, and perform model diagnostics again.

Justification for choosing the predictor as Mental Distress/Disorder:

1. The plot for Mental Distress/Disorder indicated that there is a **strong negative relationship** between mental distress and life satisfaction
2. That means that if mental distress increases, the life satisfaction will decrease.
3. The plot also shows the the there is a strong correlation since there are a lot of points near the fitted line of the regression model. This is the reason I chose Mental Distress as predictor, even though a few other predictors too displayed strong negative relation.
4. Hence, I decided to include it in the model.

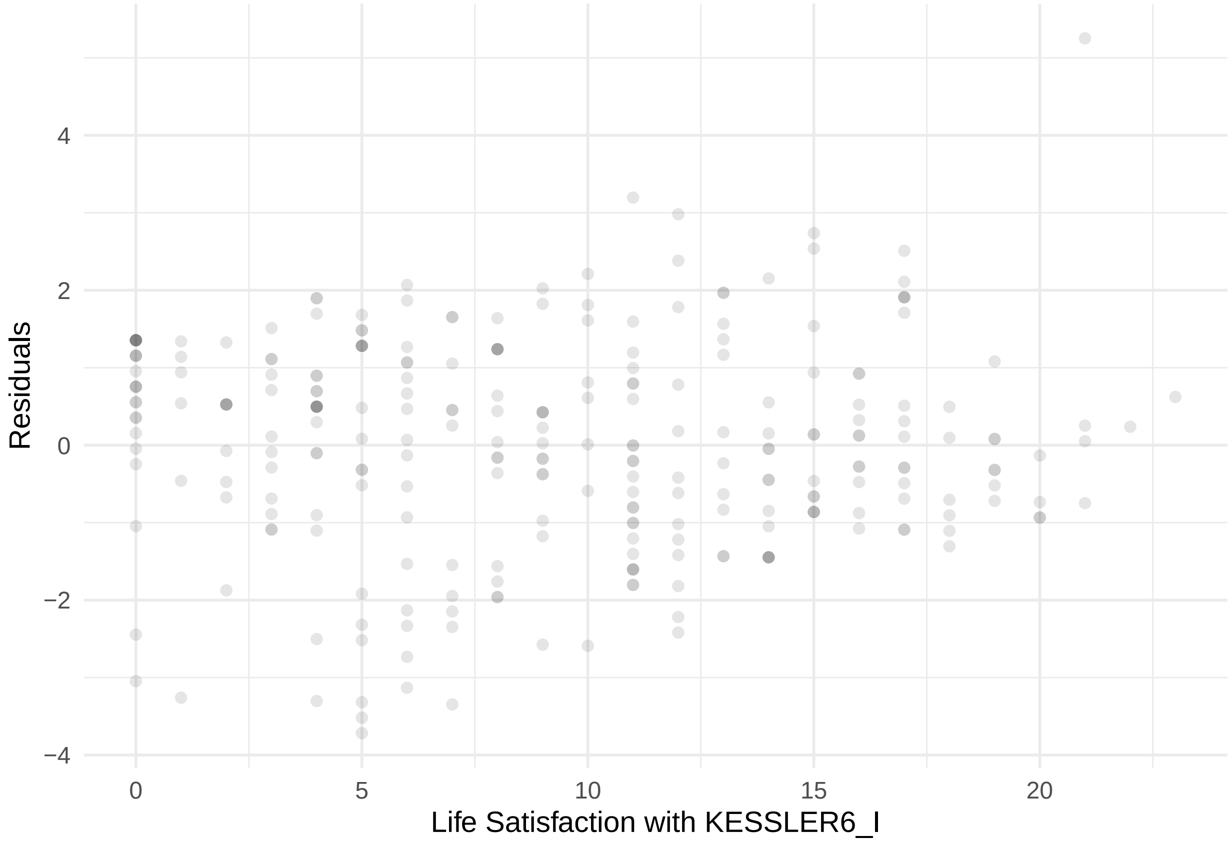
Fitting the model

```
model <- lm(LIFESAT_I ~ KESSLER6_I, data = TransPopdata)
summary(model)
```

```
##
## Call:
## lm(formula = LIFESAT_I ~ KESSLER6_I, data = TransPopdata)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7174 -0.8616  0.0807  0.9398  5.2519
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.64530    0.15443   36.56   <2e-16 ***
## KESSLER6_I  -0.18558    0.01411  -13.15   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.363 on 272 degrees of freedom
## Multiple R-squared:  0.3888, Adjusted R-squared:  0.3866
## F-statistic:   173 on 1 and 272 DF,  p-value: < 2.2e-16
```

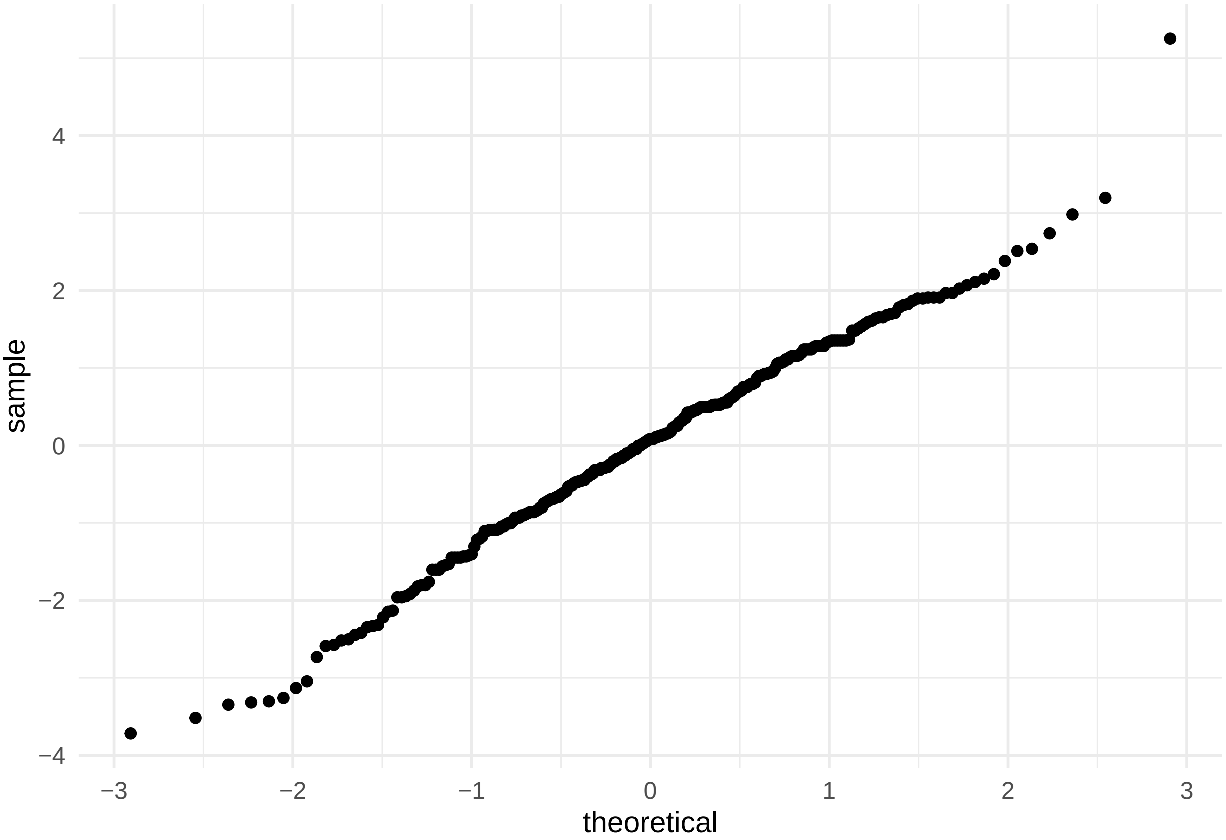
Plotting the residuals

```
TransPopdata %>%
  add_residuals(model, "resid") %>%
  ggplot(aes(x=(KESSLER6_I))) +
  geom_point(aes(y=resid), alpha=0.1) +
  labs(x="Life Satisfaction with KESSLER6_I", y="Residuals") +
  theme_minimal()
```



QQ-Plot to check if the residuals are approximately normal

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



Comment on any outliers or violations of model assumptions you notice in the residual plots.

1. In the above QQ-Plot one can observe the presence of outliers
2. Having outliers can affect the prediction accuracy of the model
3. So, the outliers need to be removed for having a better model
4. Additionally, observing the plot for residuals, one can see that there is "simple random scatter" which indicates there are no violations of model assumptions. A systematic pattern when plotted with a variable used in the model indicates a violation of model assumptions.

Finding outliers

```
# The outlier corresponds to the row with index 11,215,269:
TransPopdata_model2 <- TransPopdata
outlier <- which(abs(model$residuals) > 2.5*sd(model$residuals))
TransPopdata_model2[outlier, ]
```

```
##       STUDYID LIFESAT_I SOCIALWB_I NONAFFIRM_I NONDISCLOSURE_I HCTHREAT_I
## 11  153362805       1.2   5.800000    2.833333             5.0          4
## 215 177262357       7.0   2.266667    2.166667             3.2          5
## 269 194377738       1.0   3.933333    4.000000             4.2          4
##     KESSLER6_I EVERYDAY_I
## 11           5   1.666667
## 215         21   1.000000
## 269          5   1.000000
```

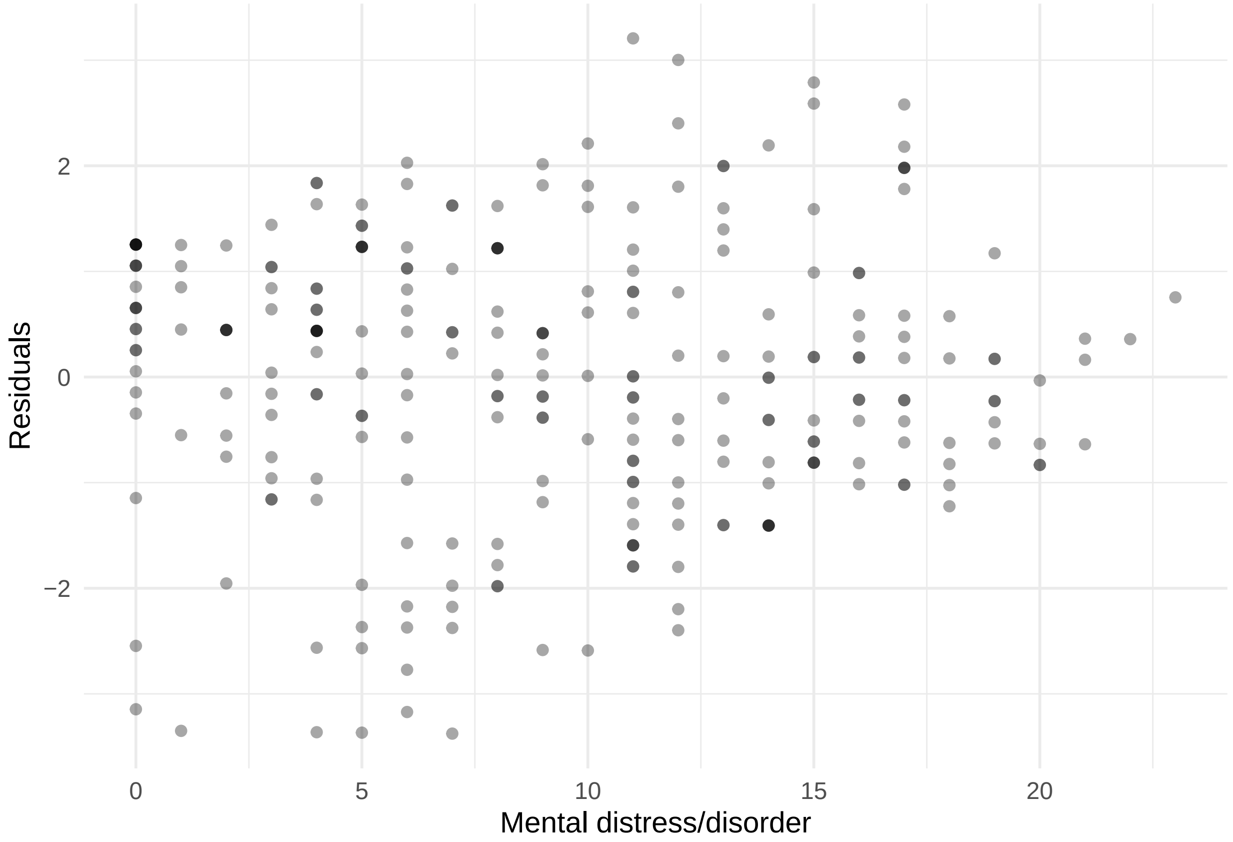
removing the outliers and refitting the model

```
# Remove the outlier
TransPopdata_model2 <- TransPopdata_model2[-outlier, ]

# Re-fit the model
model2 <- lm(LIFESAT_I ~ KESSLER6_I, data = TransPopdata_model2)

# Print the model summary
summary(model2)
```

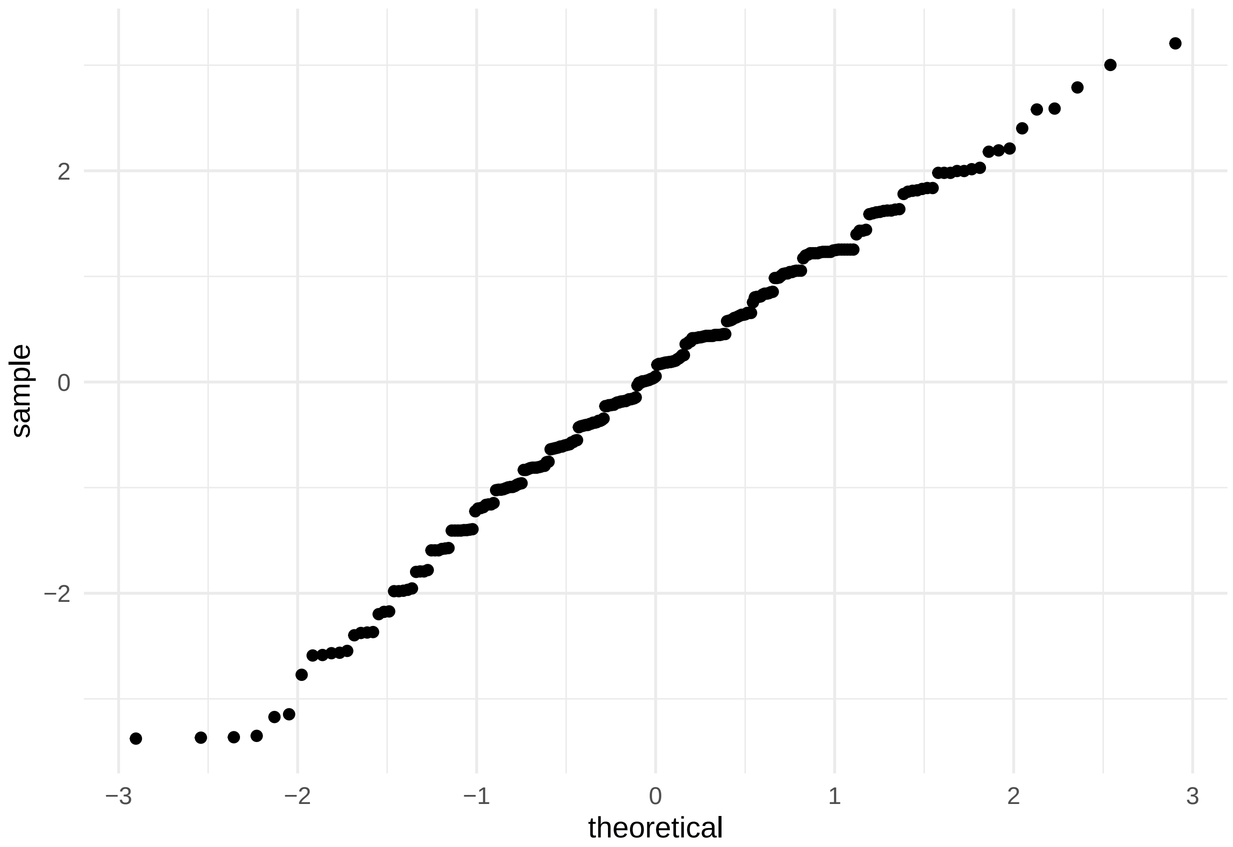
```
##
## Call:
## lm(formula = LIFESAT_I ~ KESSLER6_I, data = TransPopdata_model2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.3762 -0.8109  0.0541  0.9847  3.2064
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.74589    0.14773   38.90   <2e-16 ***
## KESSLER6_I  -0.19566    0.01352  -14.47   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.295 on 269 degrees of freedom
## Multiple R-squared:  0.4376, Adjusted R-squared:  0.4356
## F-statistic: 209.3 on 1 and 269 DF,  p-value: < 2.2e-16
```

Model Diagnostics

```
TransPopdata_model2 %>%
  add_residuals(model2, "resid") %>%
  ggplot(aes(x=(KESSLER6_I))) +
  geom_point(aes(y=resid), alpha=0.35) +
  labs(x="Mental distress/disorder", y="Residuals") +
  theme_minimal()
```



```
TransPopdata_model2 %>%
  add_residuals(model2, "resid") %>%
  ggplot(aes(sample=resid)) +
  geom_qq() +
  theme_minimal()
```



Additional comments:

1. One can observe that there are no outliers present now and the current model is better than the one with outliers

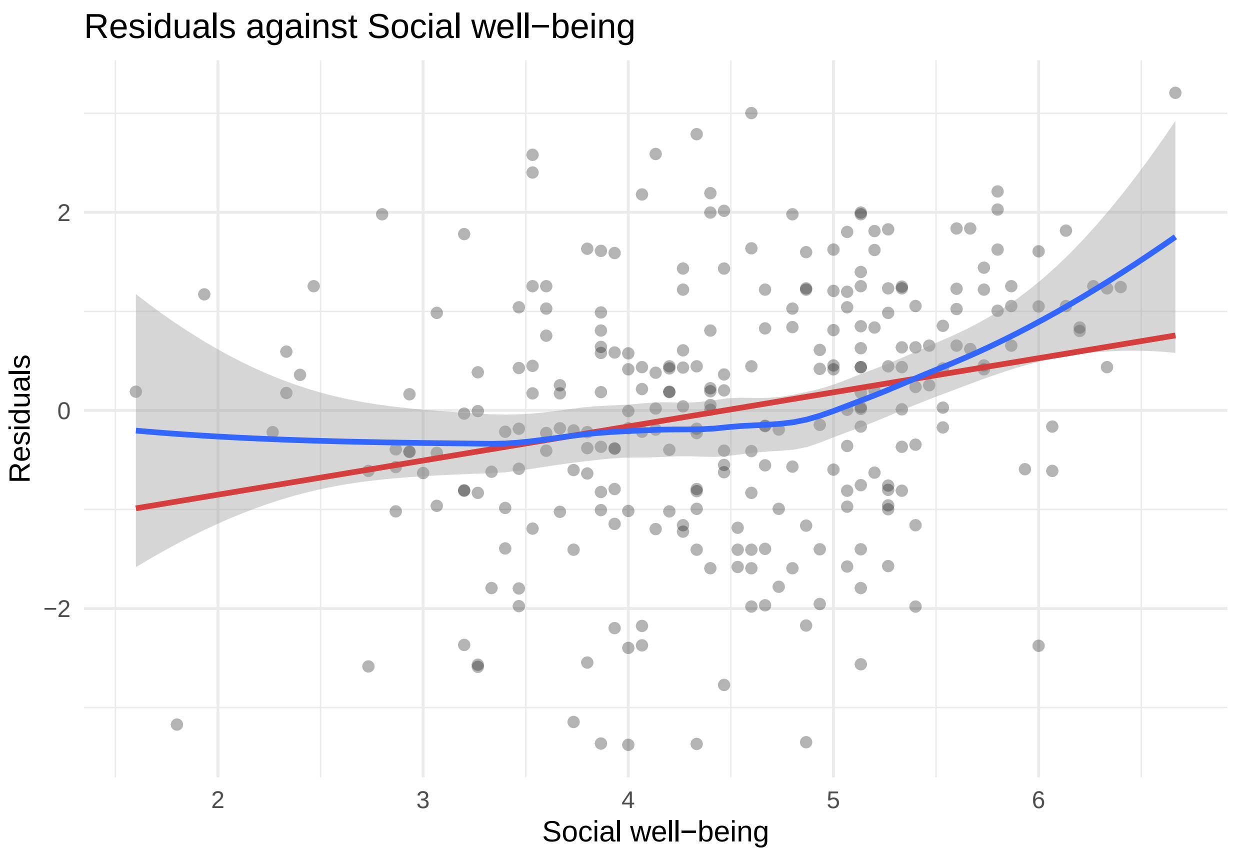
Problem 5

Use residual plots to determine if any other candidate predictors should be added to your model from Problem 2. If so, add up to one additional predictor to the model, and then perform model diagnostics on the new model.

Candidate Predictor 1: Social well-being

```
TransPopdata_model2 %>%
  add_residuals(model2, "resid") %>%
  ggplot(aes(x=SOCIALWB_I, y=resid)) +
  geom_point(alpha=0.3) +
  stat_smooth(method=lm, color="red", se=FALSE)+
  geom_smooth() +
  labs(x="Social well-being", y="Residuals") +
  ggtitle("Residuals against Social well-being") +
  theme_minimal()
```

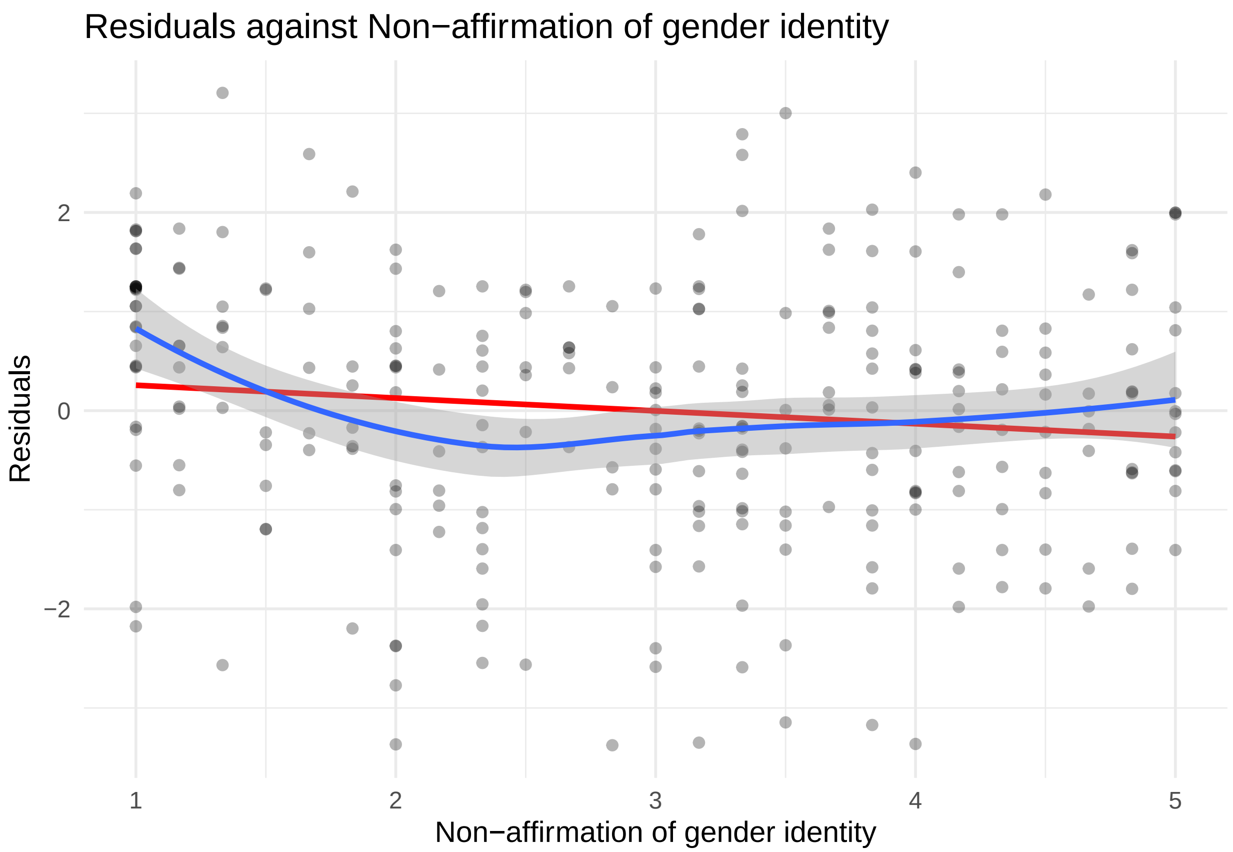
```
## ‘geom_smooth()‘ using formula = 'y ~ x'
## ‘geom_smooth()‘ using method = 'loess' and formula = 'y ~ x'
```



Candidate Predictor 2: Non-affirmation of gender identity

```
TransPopdata_model2 %>%
  add_residuals(model2, "resid") %>%
  ggplot(aes(x=NONAFFIRM_I, y=resid)) +
  geom_point(alpha=0.3) +
  stat_smooth(method=lm, color="red", se=FALSE)+
  geom_smooth() +
  labs(x="Non-affirmation of gender identity", y="Residuals") +
  ggtitle("Residuals against Non-affirmation of gender identity") +
  theme_minimal()
```

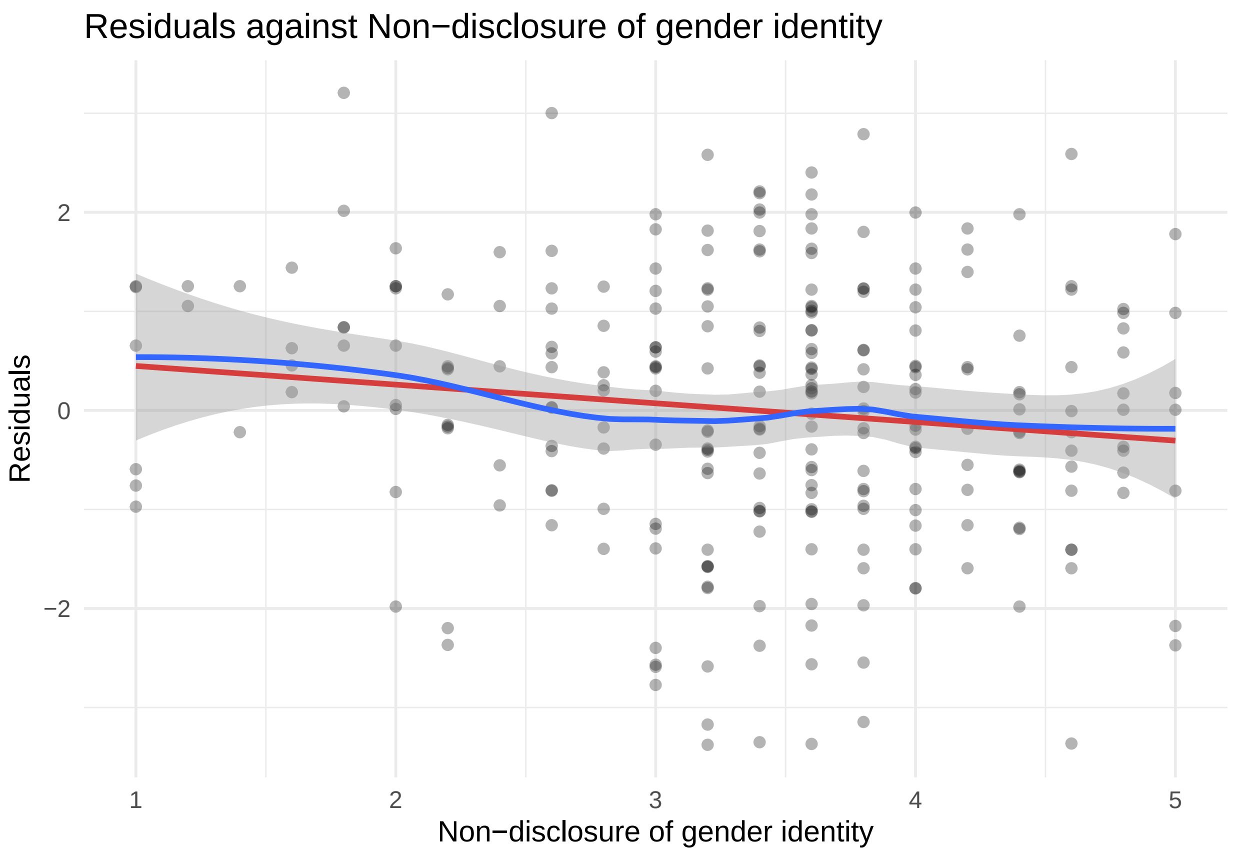
```
## ‘geom_smooth()‘ using formula = ’y ~ x’
## ‘geom_smooth()‘ using method = ’loess’ and formula = ’y ~ x’
```



Candidate Predictor 3: Non-disclosure of gender identity

```
TransPopdata_model2 %>%
  add_residuals(model2, "resid") %>%
  ggplot(aes(x=NONDISCLOSURE_I, y=resid)) +
  geom_point(alpha=0.3) +
  stat_smooth(method=lm, color="red", se=FALSE)+
  geom_smooth() +
  labs(x="Non-disclosure of gender identity", y="Residuals") +
  ggtitle("Residuals against Non-disclosure of gender identity") +
  theme_minimal()
```

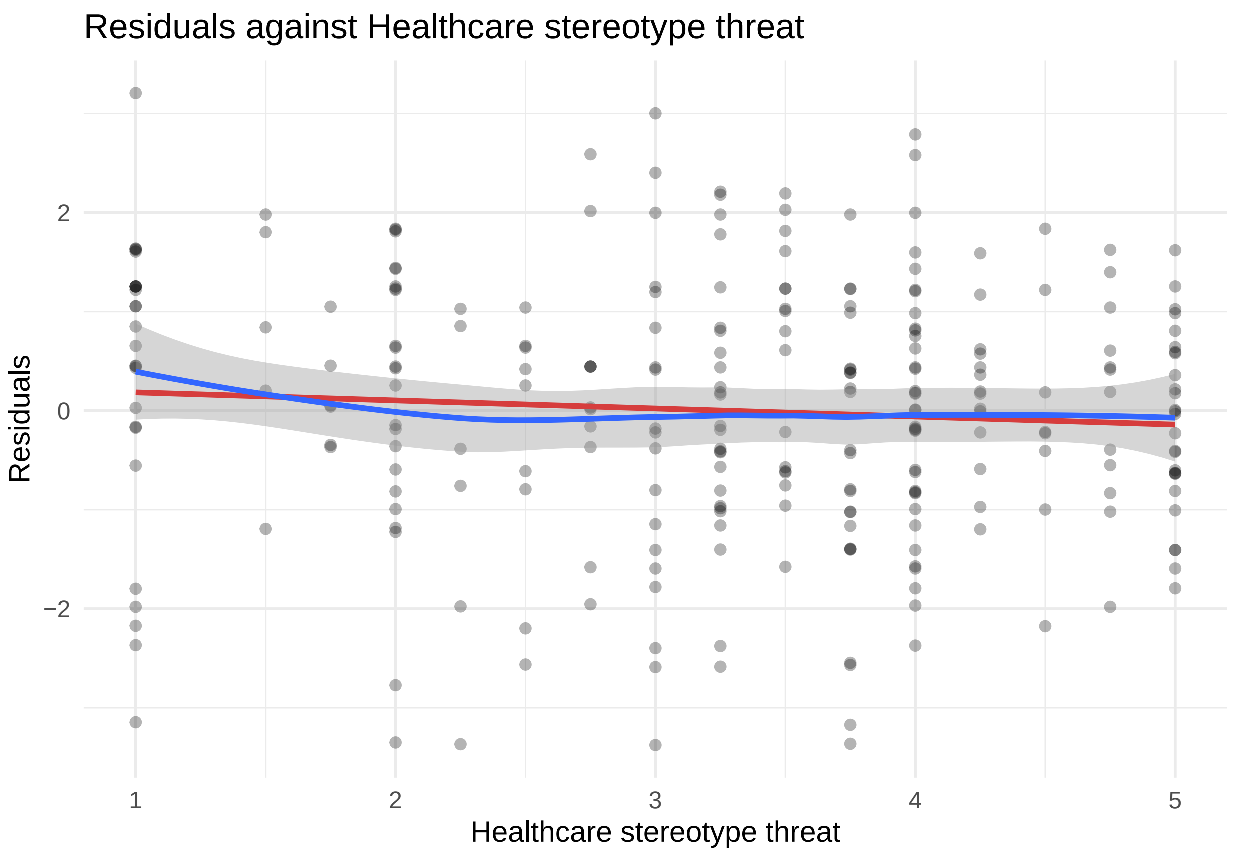
```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



Candidate Predictor 4: Healthcare stereotype threat

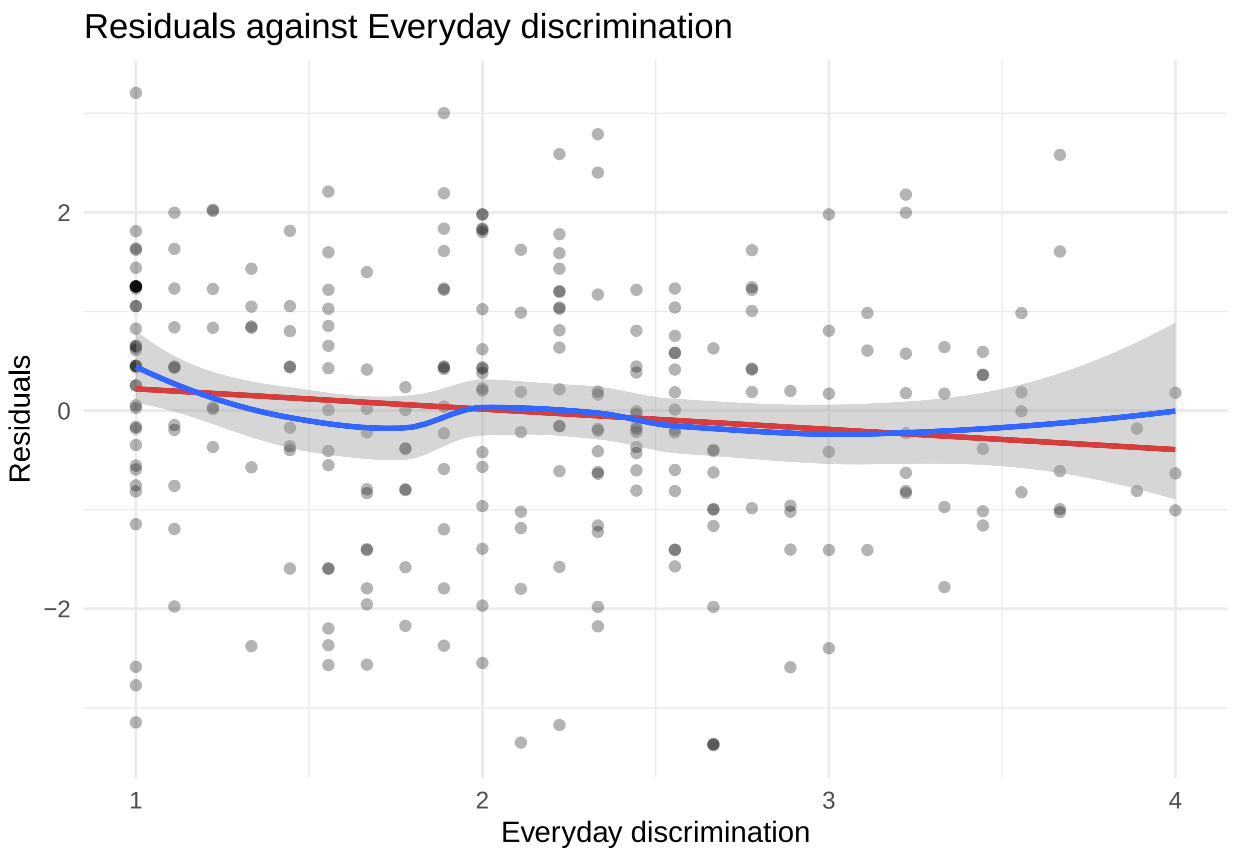
```
TransPopdata_model2 %>%
  add_residuals(model2, "resid") %>%
  ggplot(aes(x=HCTHREAT_I, y=resid)) +
  geom_point(alpha=0.3) +
  stat_smooth(method=lm, color="red", se=FALSE)+
  geom_smooth() +
  labs(x="Healthcare stereotype threat", y="Residuals") +
  ggtitle("Residuals against Healthcare stereotype threat") +
  theme_minimal()
```

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



```
TransPopdata_model2 %>%
  add_residuals(model2, "resid") %>%
  ggplot(aes(x=EVERYDAY_I, y=resid)) +
  geom_point(alpha=0.3) +
  stat_smooth(method=lm, color="red", se=FALSE)+
  geom_smooth() +
  labs(x="Everyday discrimination", y="Residuals") +
  ggtitle("Residuals against Everyday discrimination") +
  theme_minimal()
```

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



Fitting the model

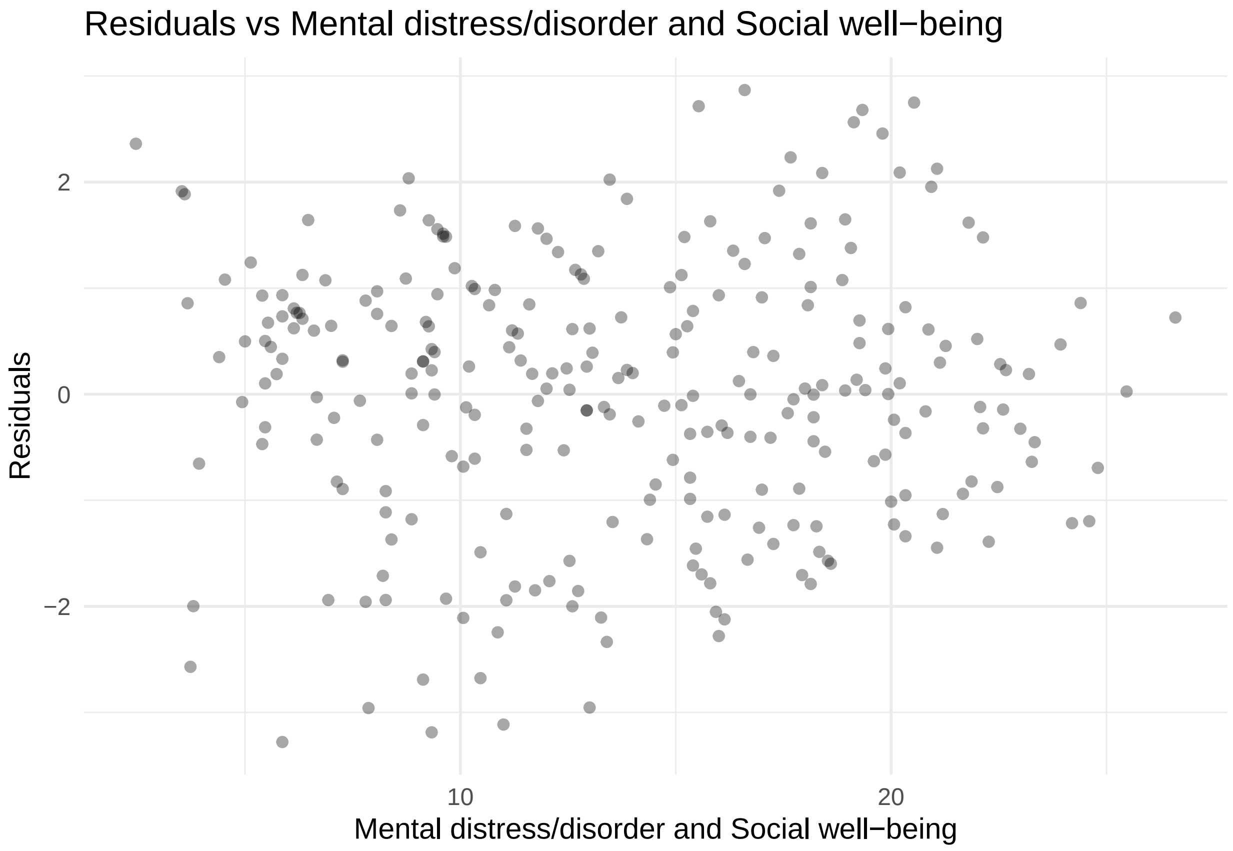
Choosing the Predictor Social well-being as the additional predictor to the model

```
model3 <- lm(LIFESAT_I ~ KESSLER6_I + SOCIALWB_I, data = TransPopdata_model2)
summary(model3)
```

```
##
## Call:
## lm(formula = LIFESAT_I ~ KESSLER6_I + SOCIALWB_I, data = TransPopdata_model2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2792 -0.8630  0.1020  0.8389  2.8680
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.60267    0.47439   7.594 5.17e-13 ***
## KESSLER6_I  -0.16684    0.01437 -11.612  < 2e-16 ***
## SOCIALWB_I   0.41987    0.08867   4.735 3.55e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.246 on 268 degrees of freedom
## Multiple R-squared:  0.4811, Adjusted R-squared:  0.4772
## F-statistic: 124.2 on 2 and 268 DF,  p-value: < 2.2e-16
```

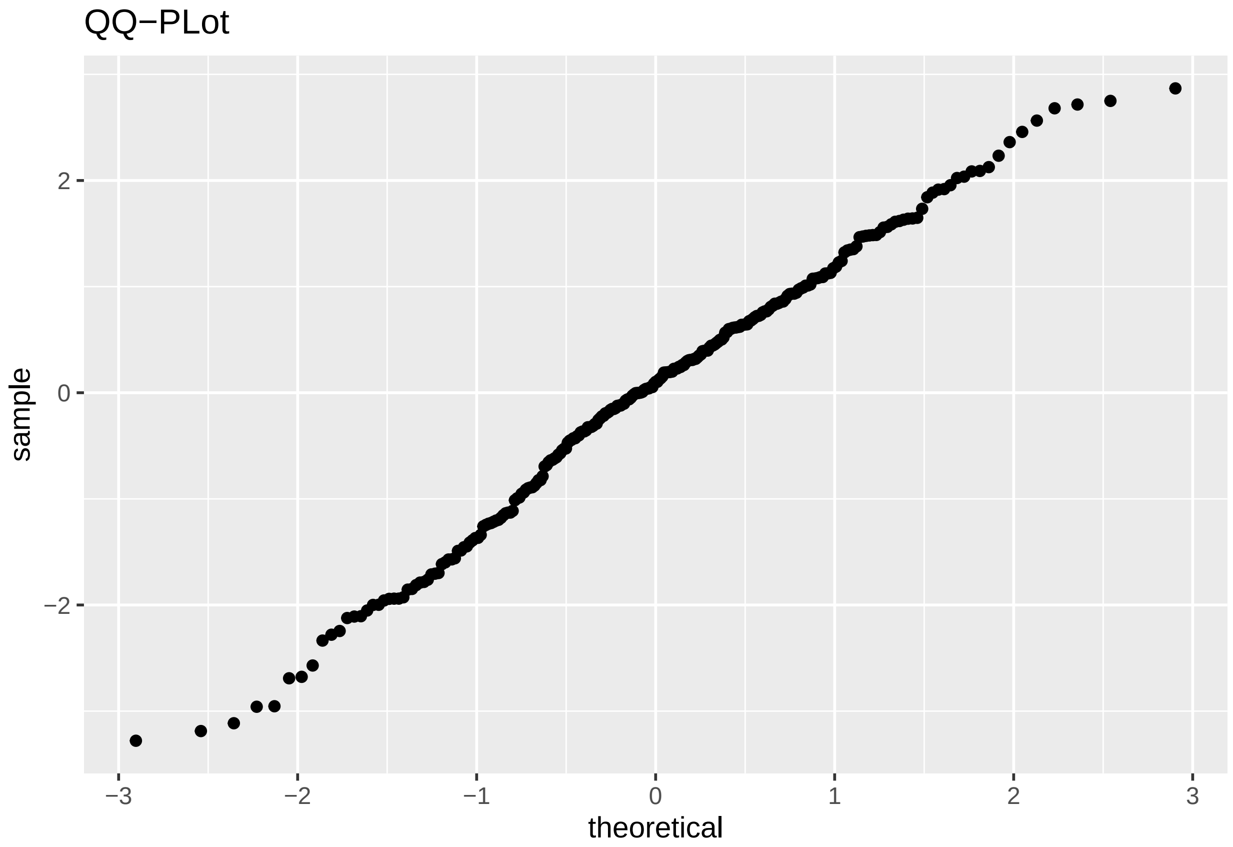
Model Diagnostics

```
TransPopdata_model2 %>%
  add_residuals(model3, "resid") %>%
  ggplot(aes(x=(KESSLER6_I + SOCIALWB_I))) +
  geom_point(aes(y=resid), alpha=0.35) +
  labs(x="Mental distress/disorder and Social well-being", y="Residuals") +
  ggtitle("Residuals vs Mental distress/disorder and Social well-being")+
  theme_minimal()
```



QQ-Plot

```
TransPopdata_model2 %>%
  add_residuals(model3, "resid") %>%
  ggplot(aes(sample=resid)) +
  geom_qq() +
  ggtitle("QQ-PLot")
```



```
theme_minimal()
```

```
## List of 97
##  $ line                      :List of 6
##   ..$ colour       : chr "black"
##   ..$ linewidth    : num 0.5
##   ..$ linetype     : num 1
##   ..$ lineend      : chr "butt"
##   ..$ arrow        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_line" "element"
##  $ rect                      :List of 5
##   ..$ fill         : chr "white"
##   ..$ colour       : chr "black"
##   ..$ linewidth    : num 0.5
##   ..$ linetype     : num 1
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_rect" "element"
##  $ text                      :List of 11
##   ..$ family       : chr ""
##   ..$ face         : chr "plain"
##   ..$ colour       : chr "black"
##   ..$ size         : num 11
##   ..$ hjust        : num 0.5
##   ..$ vjust        : num 0.5
##   ..$ angle        : num 0
##   ..$ lineheight   : num 0.9
##   ..$ margin       : ’margin’ num [1:4] 0points 0points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ title                     : NULL
##  $ aspect.ratio              : NULL
##  $ axis.title                : NULL
##  $ axis.title.x              :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : ’margin’ num [1:4] 2.75points 0points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.title.x.top          :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 0
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : ’margin’ num [1:4] 0points 0points 2.75points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.title.x.bottom       : NULL
##  $ axis.title.y              :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 1
##   ..$ angle        : num 90
##   ..$ lineheight   : NULL
##   ..$ margin       : ’margin’ num [1:4] 0points 2.75points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.title.y.left         : NULL
##  $ axis.title.y.right        :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 0
##   ..$ angle        : num -90
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 0points 0points 2.75points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text                 :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : chr "grey30"
##   ..$ size         : 'rel' num 0.8
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.x               :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 2.2points 0points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.x.top           :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 0
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 0points 2.2points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.x.bottom        : NULL
##  $ axis.text.y               :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 1
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : ’margin’ num [1:4] 0points 2.2points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.y.left          : NULL
##  $ axis.text.y.right         :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 0
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : ’margin’ num [1:4] 0points 0points 0points 2.2points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.ticks                : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ axis.ticks.x              : NULL
##  $ axis.ticks.x.top          : NULL
##  $ axis.ticks.x.bottom       : NULL
##  $ axis.ticks.y              : NULL
##  $ axis.ticks.y.left         : NULL
##  $ axis.ticks.y.right        : NULL
##  $ axis.ticks.length         : ’simpleUnit’ num 2.75points
##   ..- attr(*, "unit")= int 8
##  $ axis.ticks.length.x       : NULL
##  $ axis.ticks.length.x.top   : NULL
##  $ axis.ticks.length.x.bottom: NULL
##  $ axis.ticks.length.y       : NULL
##  $ axis.ticks.length.y.left  : NULL
##  $ axis.ticks.length.y.right : NULL
##  $ axis.line                 : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ axis.line.x               : NULL
##  $ axis.line.x.top           : NULL
##  $ axis.line.x.bottom        : NULL
##  $ axis.line.y               : NULL
##  $ axis.line.y.left          : NULL
##  $ axis.line.y.right         : NULL
##  $ legend.background         : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ legend.margin             : 'margin' num [1:4] 5.5points 5.5points 5.5points 5.5points
##   ..- attr(*, "unit")= int 8
##  $ legend.spacing            : 'simpleUnit' num 11points
##   ..- attr(*, "unit")= int 8
##  $ legend.spacing.x          : NULL
##  $ legend.spacing.y          : NULL
##  $ legend.key                : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ legend.key.size           : 'simpleUnit' num 1.2lines
##   ..- attr(*, "unit")= int 3
##  $ legend.key.height         : NULL
##  $ legend.key.width          : NULL
##  $ legend.text               :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : 'rel' num 0.8
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ legend.text.align         : NULL
##  $ legend.title              :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 0
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ legend.title.align        : NULL
##  $ legend.position           : chr "right"
##  $ legend.direction          : NULL
##  $ legend.justification      : chr "center"
##  $ legend.box                : NULL
##  $ legend.box.just           : NULL
##  $ legend.box.margin         : 'margin' num [1:4] 0cm 0cm 0cm 0cm
##   ..- attr(*, "unit")= int 1
##  $ legend.box.background     : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ legend.box.spacing        : 'simpleUnit' num 11points
##   ..- attr(*, "unit")= int 8
##  $ panel.background          : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ panel.border              : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ panel.spacing             : ’simpleUnit’ num 5.5points
##   ..- attr(*, "unit")= int 8
##  $ panel.spacing.x           : NULL
##  $ panel.spacing.y           : NULL
##  $ panel.grid                :List of 6
##   ..$ colour       : chr "grey92"
##   ..$ linewidth    : NULL
##   ..$ linetype     : NULL
##   ..$ lineend      : NULL
##   ..$ arrow        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_line" "element"
##  $ panel.grid.major          : NULL
##  $ panel.grid.minor          :List of 6
##   ..$ colour       : NULL
##   ..$ linewidth    : ’rel’ num 0.5
##   ..$ linetype     : NULL
##   ..$ lineend      : NULL
##   ..$ arrow        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_line" "element"
##  $ panel.grid.major.x        : NULL
##  $ panel.grid.major.y        : NULL
##  $ panel.grid.minor.x        : NULL
##  $ panel.grid.minor.y        : NULL
##  $ panel.ontop               : logi FALSE
##  $ plot.background           : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ plot.title                :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : ’rel’ num 1.2
##   ..$ hjust        : num 0
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : ’margin’ num [1:4] 0points 0points 5.5points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ plot.title.position       : chr "panel"
##  $ plot.subtitle             :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 0
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : ’margin’ num [1:4] 0points 0points 5.5points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ plot.caption              :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : ’rel’ num 0.8
##   ..$ hjust        : num 1
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : ’margin’ num [1:4] 5.5points 0points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ plot.caption.position     : chr "panel"
##  $ plot.tag                  :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : ’rel’ num 1.2
##   ..$ hjust        : num 0.5
##   ..$ vjust        : num 0.5
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ plot.tag.position         : chr "topleft"
##  $ plot.margin               : ’margin’ num [1:4] 5.5points 5.5points 5.5points 5.5points
##   ..- attr(*, "unit")= int 8
##  $ strip.background          : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ strip.background.x        : NULL
##  $ strip.background.y        : NULL
##  $ strip.clip                : chr "inherit"
##  $ strip.placement           : chr "inside"
##  $ strip.text                :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : chr "grey10"
##   ..$ size         : ’rel’ num 0.8
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : ’margin’ num [1:4] 4.4points 4.4points 4.4points 4.4points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ strip.text.x              : NULL
##  $ strip.text.x.bottom       : NULL
##  $ strip.text.x.top          : NULL
##  $ strip.text.y              :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : num -90
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ strip.text.y.left         :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : num 90
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ strip.text.y.right        : NULL
##  $ strip.switch.pad.grid     : ’simpleUnit’ num 2.75points
##     ..- attr(*, "unit")= int 8
##  $ strip.switch.pad.wrap     : ’simpleUnit’ num 2.75points
##     ..- attr(*, "unit")= int 8
##  - attr(*, "class")= chr [1:2] "theme" "gg"
##  - attr(*, "complete")= logi TRUE
##  - attr(*, "validate")= logi TRUE
```