STATS 485 Paper 2 Appendix, Version 2

Caroline Moy

2025-02-19

This appendix contains the calculations for the paper "Is Overconfidence a Gender Issue?" First the necessary packages and data will be loaded in, followed by exploratory data analysis, fitting of a linear regression model to predict overconfidence using intelligence theory and attention experimental condition and then using the results to motivate adding gender into the next model. The mean squared error of each model is then computed and finally an ANOVA test between the two models is performed, to see if adding gender is improving model fit in a meaningful way. With version 2, a hold out set was provided, so the mean-squared error is computed for the model with and without gender, to see the generalizability of these two models. In addition, a Wilcoxon signed ranks test is performed to see if the residuals between these two models are significantly different, does one model systematically give larger or smaller residuals on the hold out set?

Necessary Libraries and Datasets

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- 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(readr)
library(splines)
library(caret)
```

```
## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(MASS)
```

```
##
## Attaching package: 'MASS'
##
## The following object is masked from 'package:dplyr':
```

```
##
##     select
```

```
attention = read_csv("http://dept.stat.lsa.umich.edu/~bbh/s485/data/emdstudy3-small-nogender.csv")
```

```
## Rows: 70 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): attn_to
## dbl (3): intel_theory, ActPerc, EstPerc
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
attention_gender = read_csv("http://dept.stat.lsa.umich.edu/~bbh/s485/data/emdstudy3-small.csv")
```

```
## Rows: 70 Columns: 5
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): gender, attn_to
## dbl (3): intel_theory, ActPerc, EstPerc
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
hold_out_set = read_csv("http://dept.stat.lsa.umich.edu/~bbh/s485/data/emdstudy3-holdout.csv")
```

```
## Rows: 34 Columns: 5
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): gender, attn_to
## dbl (3): intel_theory, ActPerc, EstPerc
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Exploratory Data Analysis

Creating overconfidence variable.

```
attention_gender = attention_gender %>%
  mutate(overconfidence = EstPerc - ActPerc)
```

Checking for NA in dataset.

```
sum(is.na(attention_gender))
```

```
## [1] 0
```

Finding overall mean of overconfidence.

```
mean(attention_gender$overconfidence)
```

```
## [1] 12.15214
```

Finding top 10 scorers of exam and individuals with top 10 estimated percentiles.

```
attention_gender %>%
  arrange(desc(ActPerc)) %>%
```

```
  head(10) %>%
  group_by(gender) %>%
  summarize(gender_count = n())
```

```
## # A tibble: 1 x 2
##   gender gender_count
##   <chr>         <int>
## 1 W                10
```

```
attention_gender %>%
  arrange(desc(EstPerc)) %>%
  head(10) %>%
  group_by(gender) %>%
  summarize(gender_count = n())
```

```
## # A tibble: 2 x 2
##   gender gender_count
##   <chr>         <int>
## 1 M                 6
## 2 W                 4
```

Gender differences in Overconfidence.

```
gender_diff = attention_gender %>%
  group_by(gender) %>%
  mutate('Gender' = gender) %>%
  summarize(`Average Percent` = mean(ActPerc),
            `Standard Deviation` = sqrt(var(ActPerc)))

knitr::kable(gender_diff,
             caption = "Overconfidence Stratified by Gender and Attention Treatment",
             "simple",
             digits = 3)
```

Table 1: Overconfidence Stratified by Gender and Attention Treatment

gender	Average Percent	Standard Deviation
M	42.259	27.919
W	54.726	28.713

Overconfidence based on experimental condition and gender.

```
attention_gender %>%
  group_by(gender, attn_to) %>%
  summarize(avg_overconfidence = mean(overconfidence),
            std_overconfidence = sqrt(var(overconfidence)))
```

```
## `summarise()` has grouped output by 'gender'. You can override using the
## `.groups` argument.
```

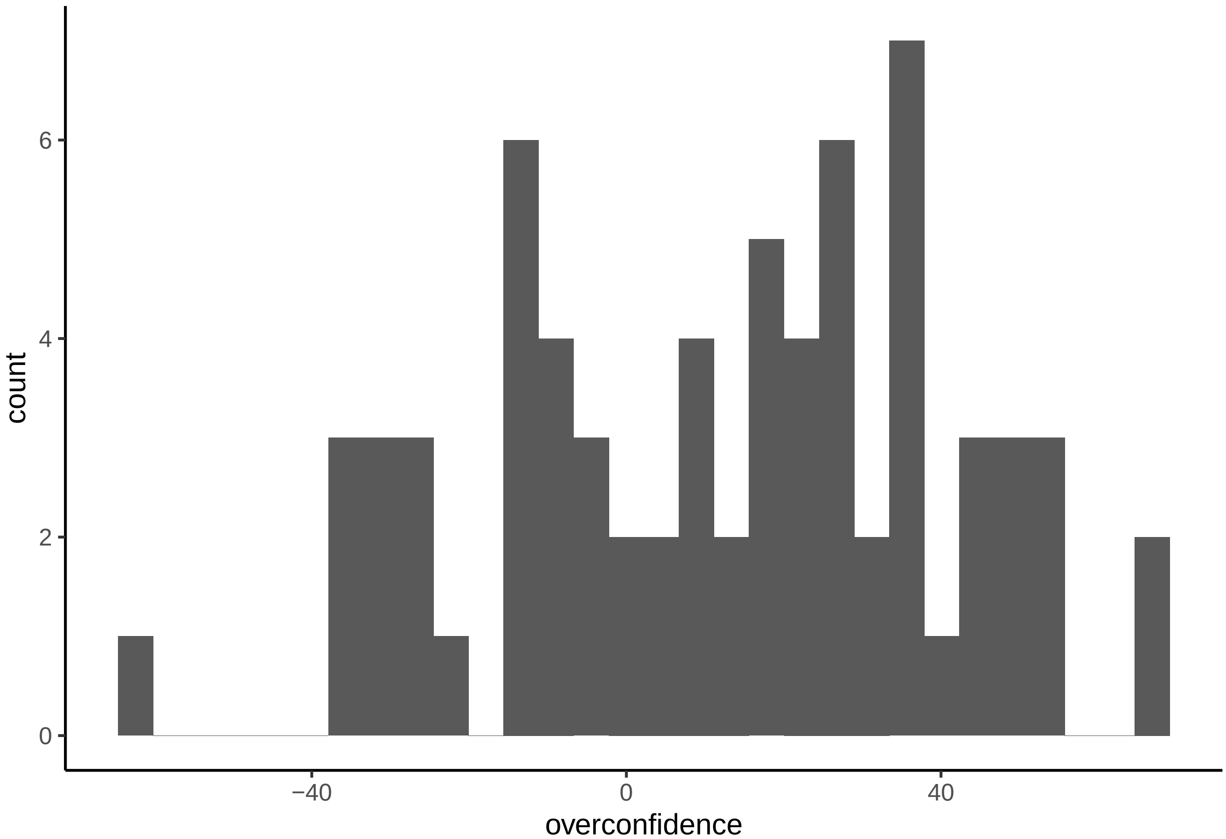
```
## # A tibble: 4 x 4
## # Groups:   gender [2]
##   gender attn_to   avg_overconfidence std_overconfidence
##   <chr>  <chr>                  <dbl>              <dbl>
```

```
## 1 M      easyprobs              26.1              26.5
## 2 M      hardprobs              22.3              28.3
## 3 W      easyprobs               2.40             22.5
## 4 W      hardprobs               3.24             30.0
```

Graph of Overconfidence.

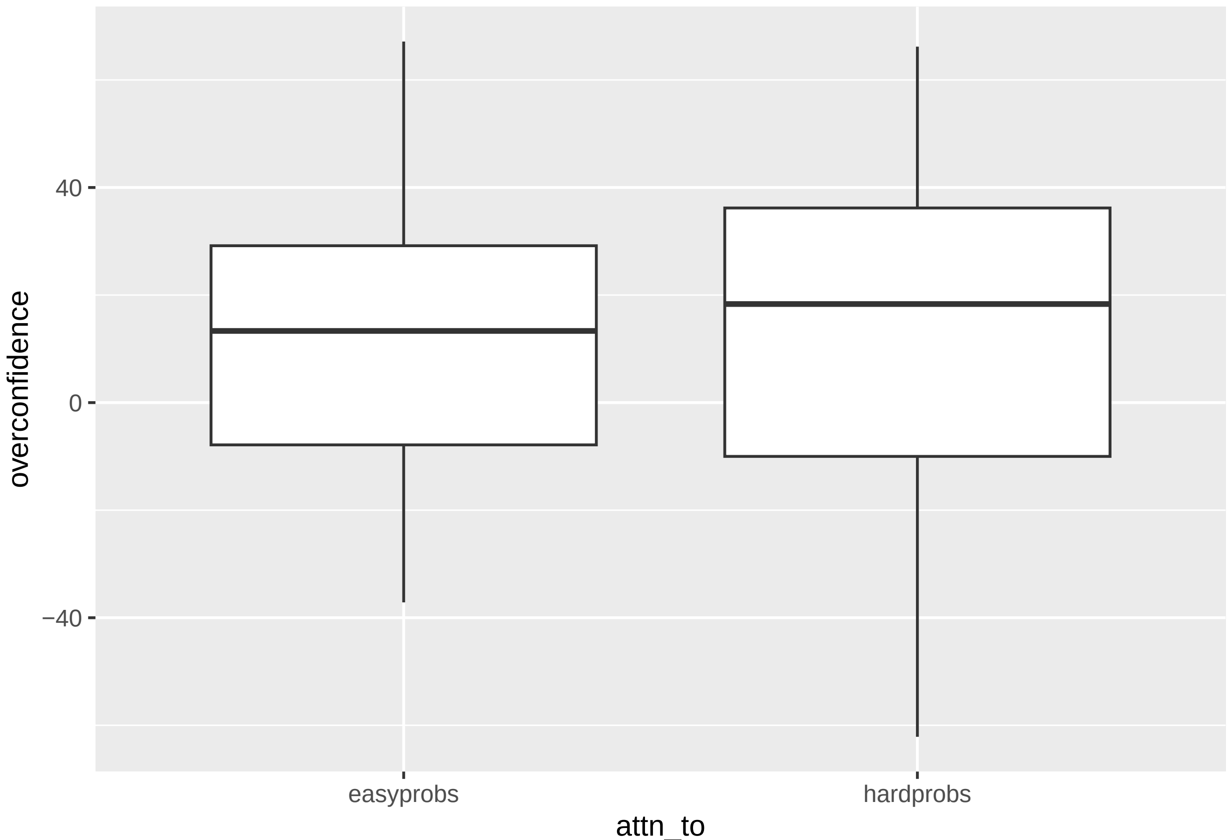
```
attention_gender %>%
  ggplot(aes(x = overconfidence)) +
  geom_histogram() +
  theme_classic()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

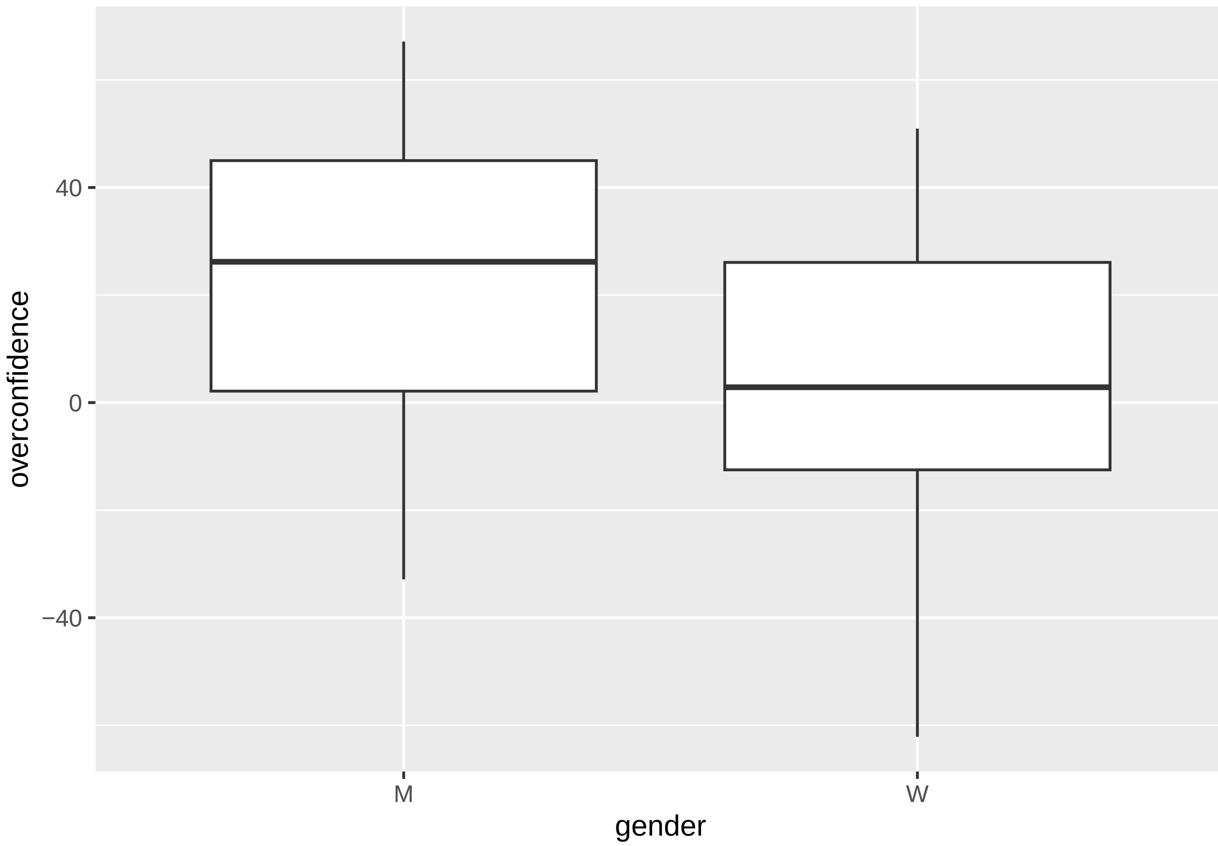


Variety of Graphs Studying Varibles' Relationships.

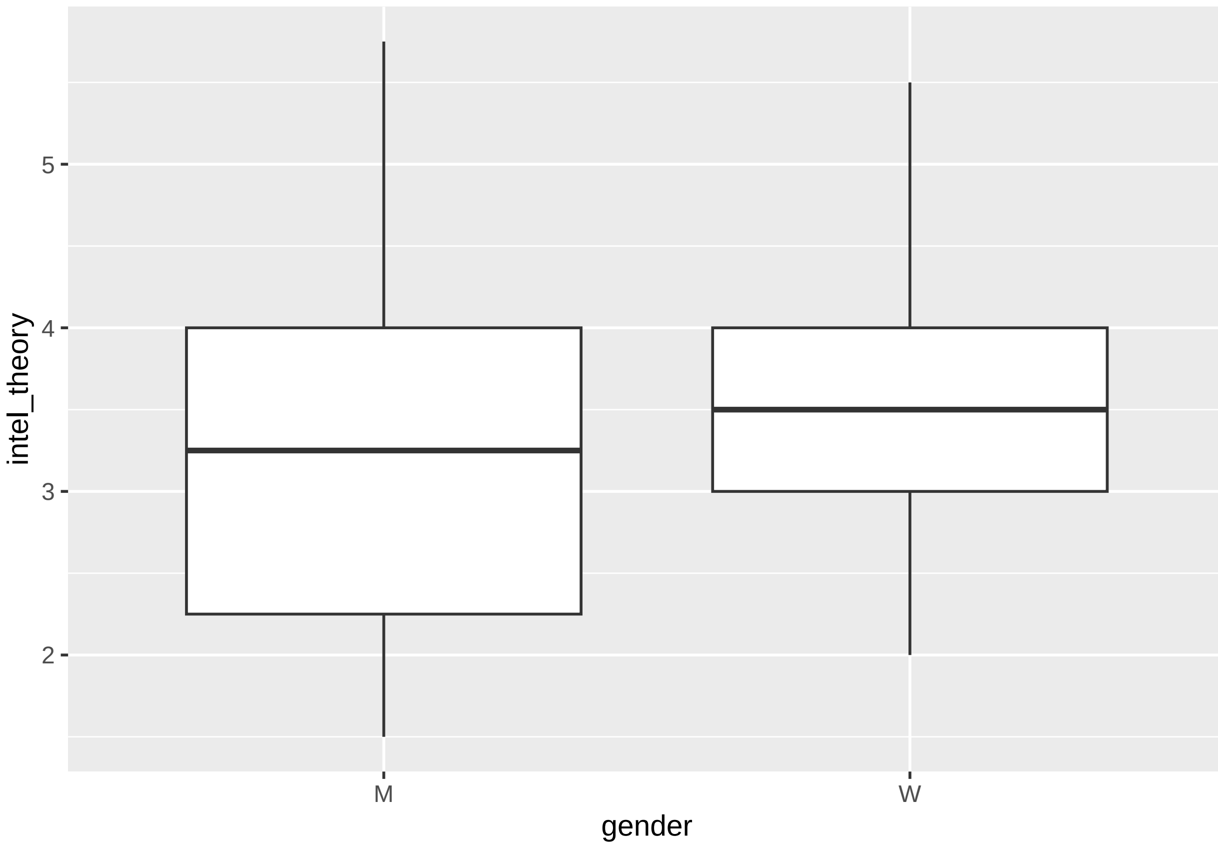
```
attention_gender %>%
  ggplot(aes(x = attn_to, y = overconfidence)) +
  geom_boxplot()
```




```r
attention_gender %>%
  ggplot(aes(x = gender, y = overconfidence)) +
  geom_boxplot()
```

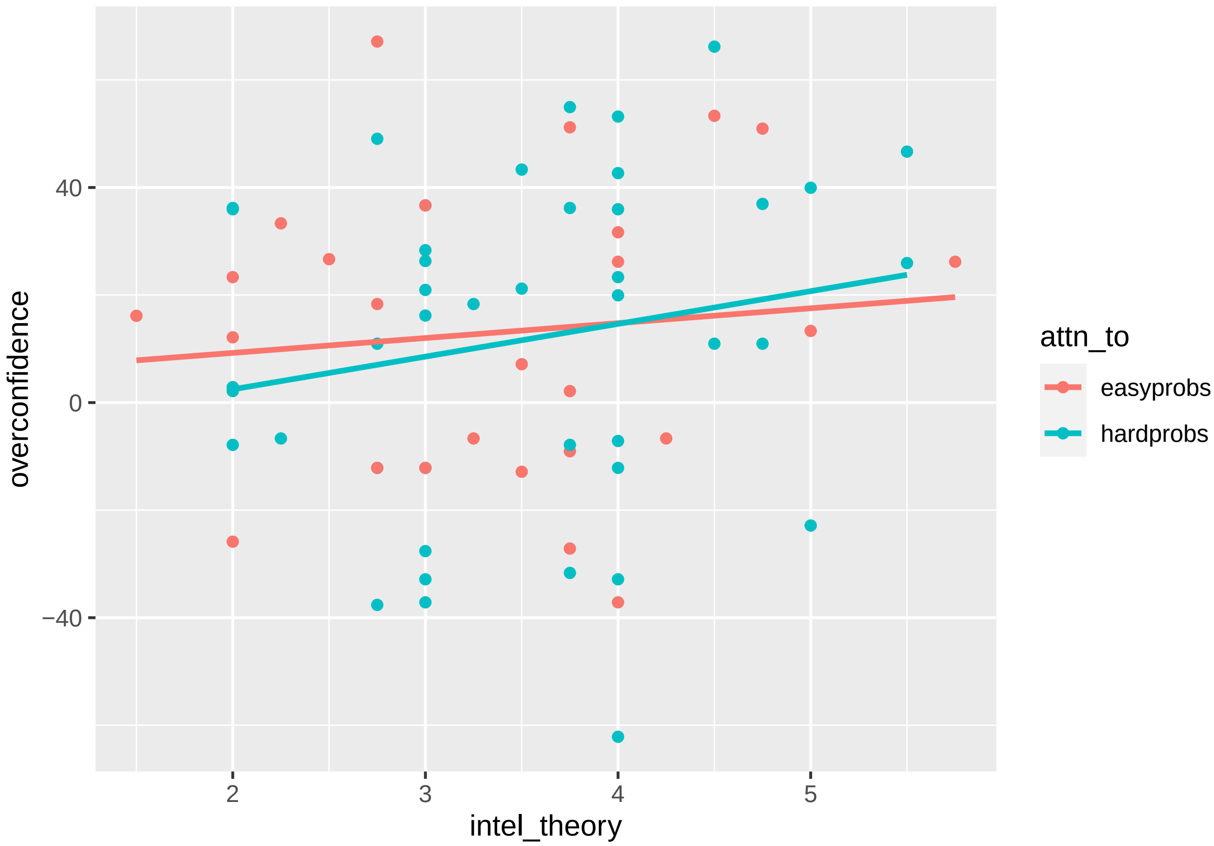



```
attention_gender %>%
  ggplot(aes(x = gender, y = intel_theory)) +
  geom_boxplot()
```



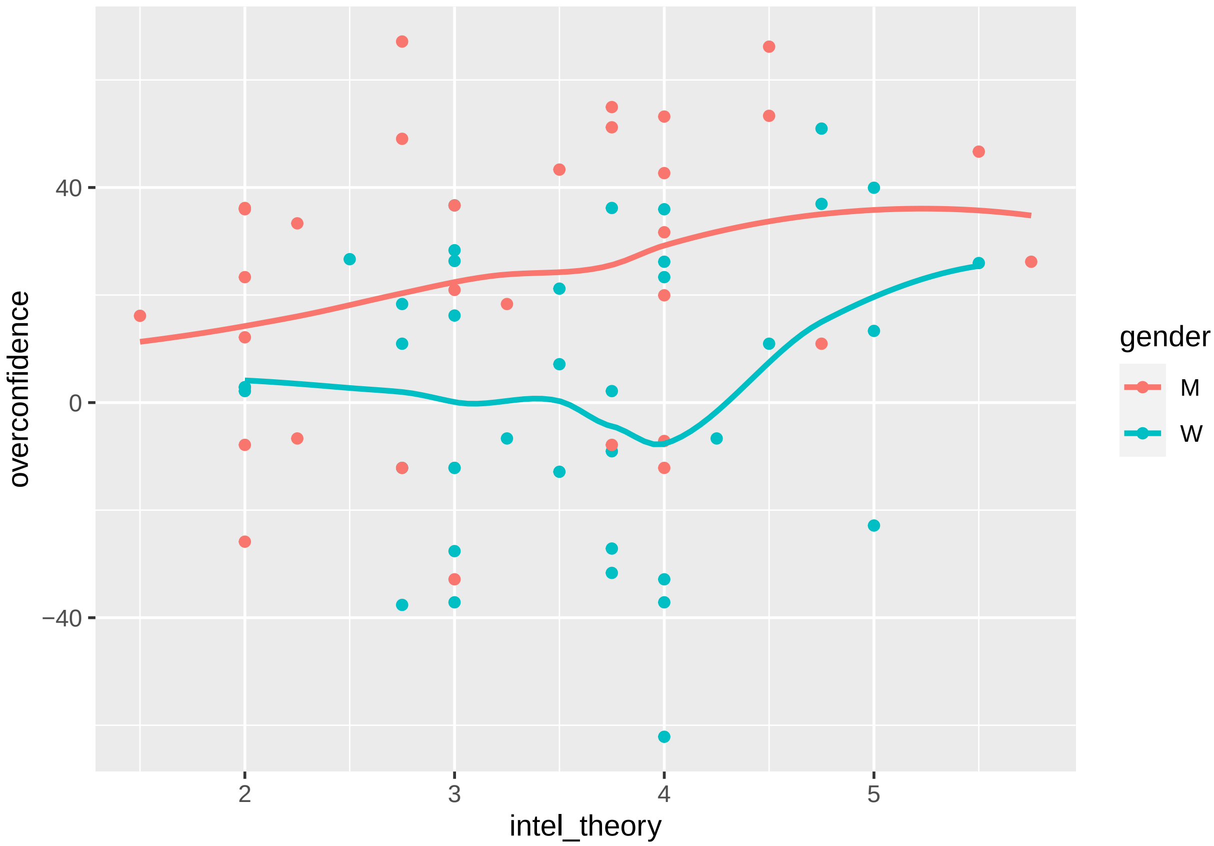

```
attention_gender %>%
  ggplot(aes(x = intel_theory, y = overconfidence, color = attn_to)) +
  geom_point() +
  geom_smooth(method = 'lm', se = FALSE)
```

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



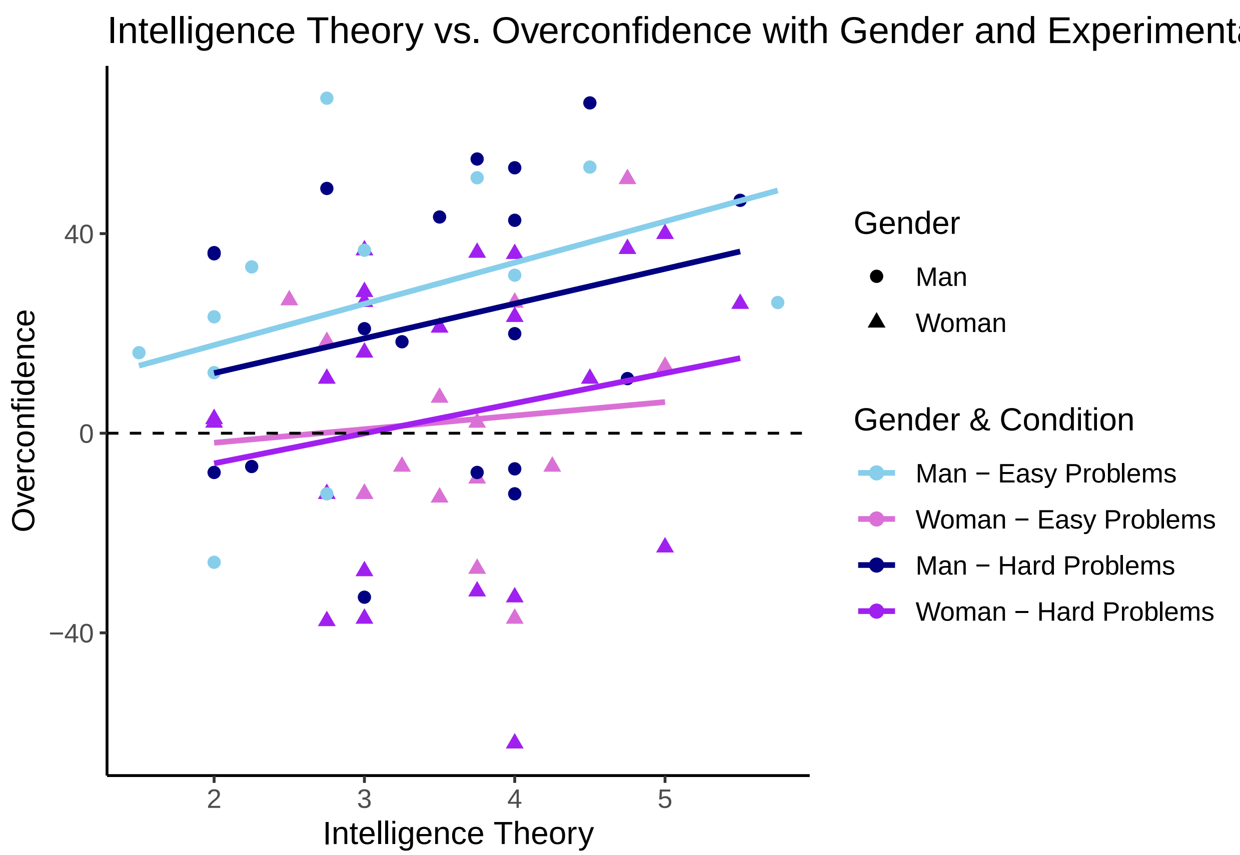

```
attention_gender %>%
  ggplot(aes(x = intel_theory, y = overconfidence, color = gender)) +
  geom_point() +
  geom_smooth(se = FALSE)
```

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



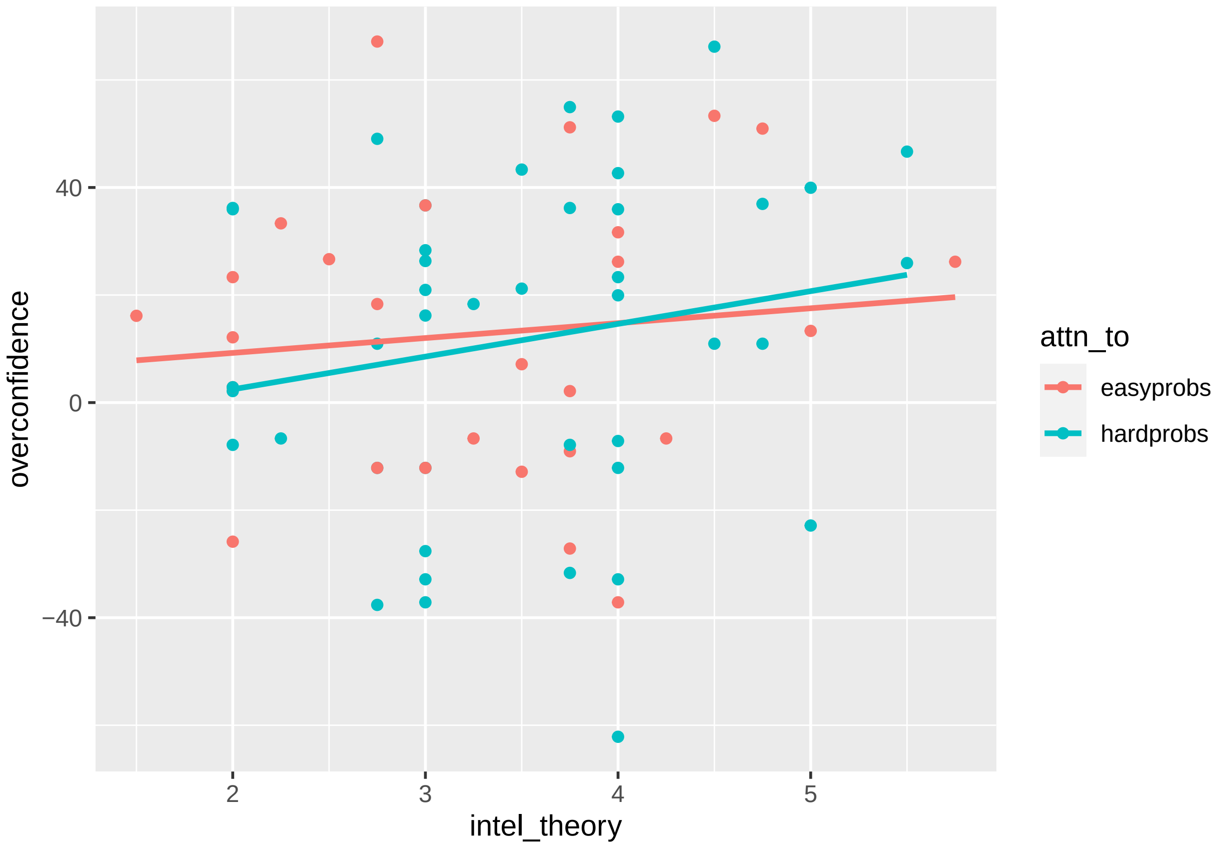

```r
attention_gender %>%
  ggplot(aes(x = intel_theory, y = overconfidence, color = interaction(gender, attn_to), shape = gender
  geom_point(size = 2) +
  geom_smooth(aes(color = interaction(gender, attn_to)), method = 'lm', se = FALSE) +
  geom_hline(yintercept = 0, linetype = 'dashed') +
  theme_classic() +
  labs(x = "Intelligence Theory", y = 'Overconfidence', color = "Gender & Condition", shape = "Gender")
  scale_color_manual(values = c("M.hardprobs" = "navy", "W.hardprobs" = "purple",
                              "M.easyprobs" = "skyblue", "W.easyprobs" = "orchid"),
                     labels = c("M.hardprobs" = "Man - Hard Problems",
                              "W.hardprobs" = "Woman - Hard Problems",
                              "M.easyprobs" = "Man - Easy Problems",
                              "W.easyprobs" = "Woman - Easy Problems")) +
  scale_shape_manual(values = c("M" = 16, "W" = 17),
                     labels = c("M" = "Man", "W" = "Woman")) +
  ggtitle('Intelligence Theory vs. Overconfidence with Gender and Experimental Condition') +
  theme(text=element_text(size=12))
```

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




```
attention_gender %>%
  ggplot(aes(x = intel_theory, y = overconfidence, color = attn_to)) +
  geom_point() +
  geom_smooth(method = 'lm', se = FALSE)
```

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



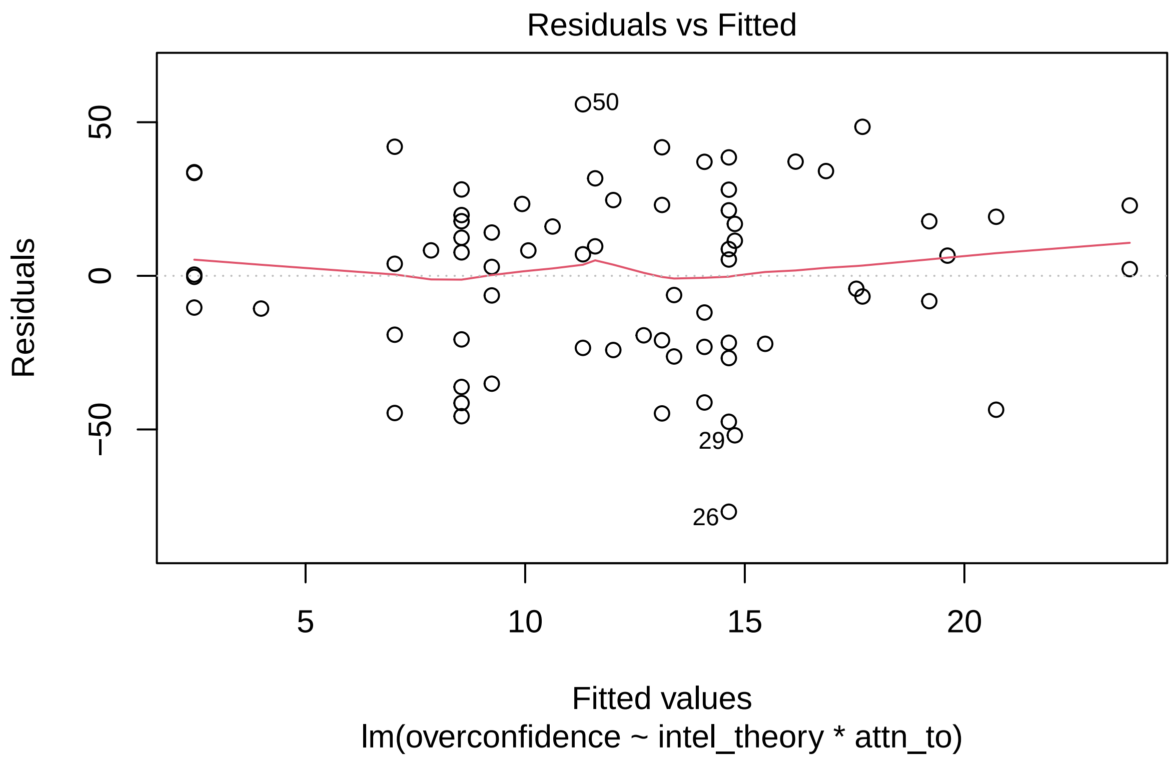
Linear Model Fitting

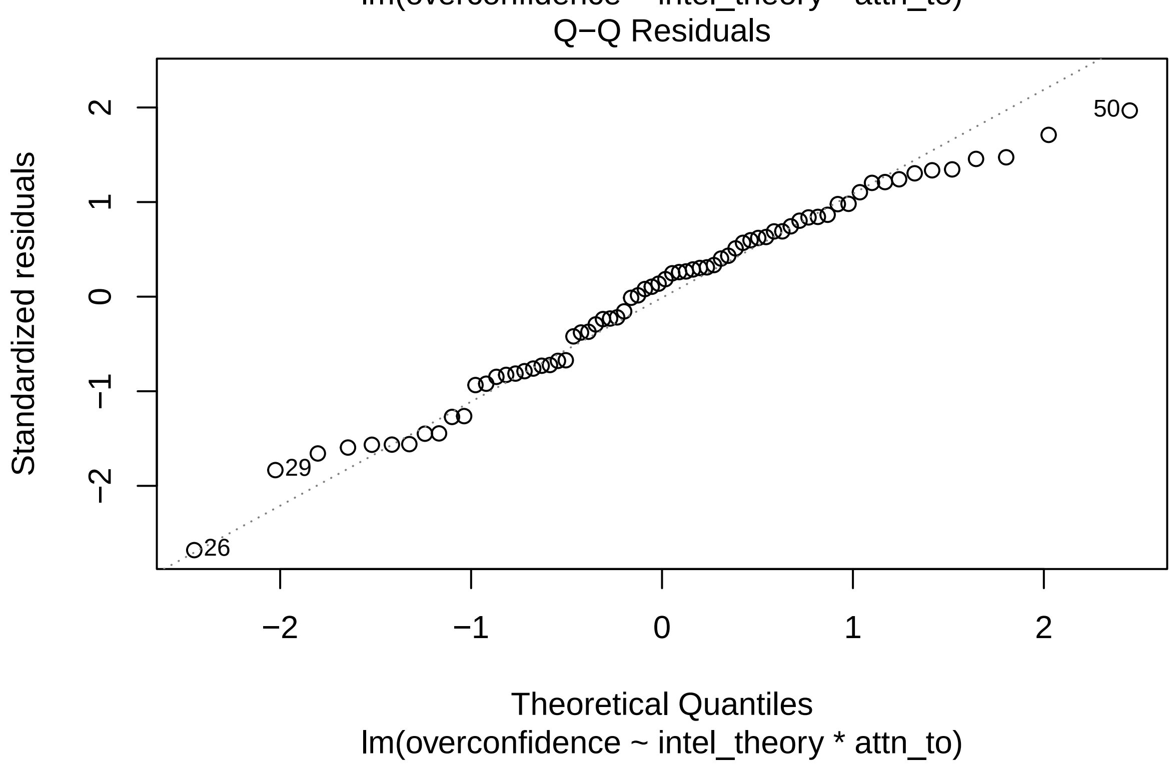
Model without Gender, with interaction terms.

```
lin_int_mod = lm(overconfidence ~ intel_theory * attn_to, data = attention_gender)
summary(lin_int_mod)
```

```
##
## Call:
## lm(formula = overconfidence ~ intel_theory * attn_to, data = attention_gender)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -76.777 -21.576   4.617  20.930  55.825
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     3.706     19.005   0.195    0.846
## intel_theory                    2.767      5.449   0.508    0.613
## attn_tohardprobs              -13.411     25.674  -0.522    0.603
## intel_theory:attn_tohardprobs   3.319      7.229   0.459    0.648
##
## Residual standard error: 29.09 on 66 degrees of freedom
## Multiple R-squared:  0.02841,    Adjusted R-squared:  -0.01575
## F-statistic: 0.6434 on 3 and 66 DF,  p-value: 0.5899
```

```
for(i in 1:2){
  plot(lin_int_mod, which=i)
}
```

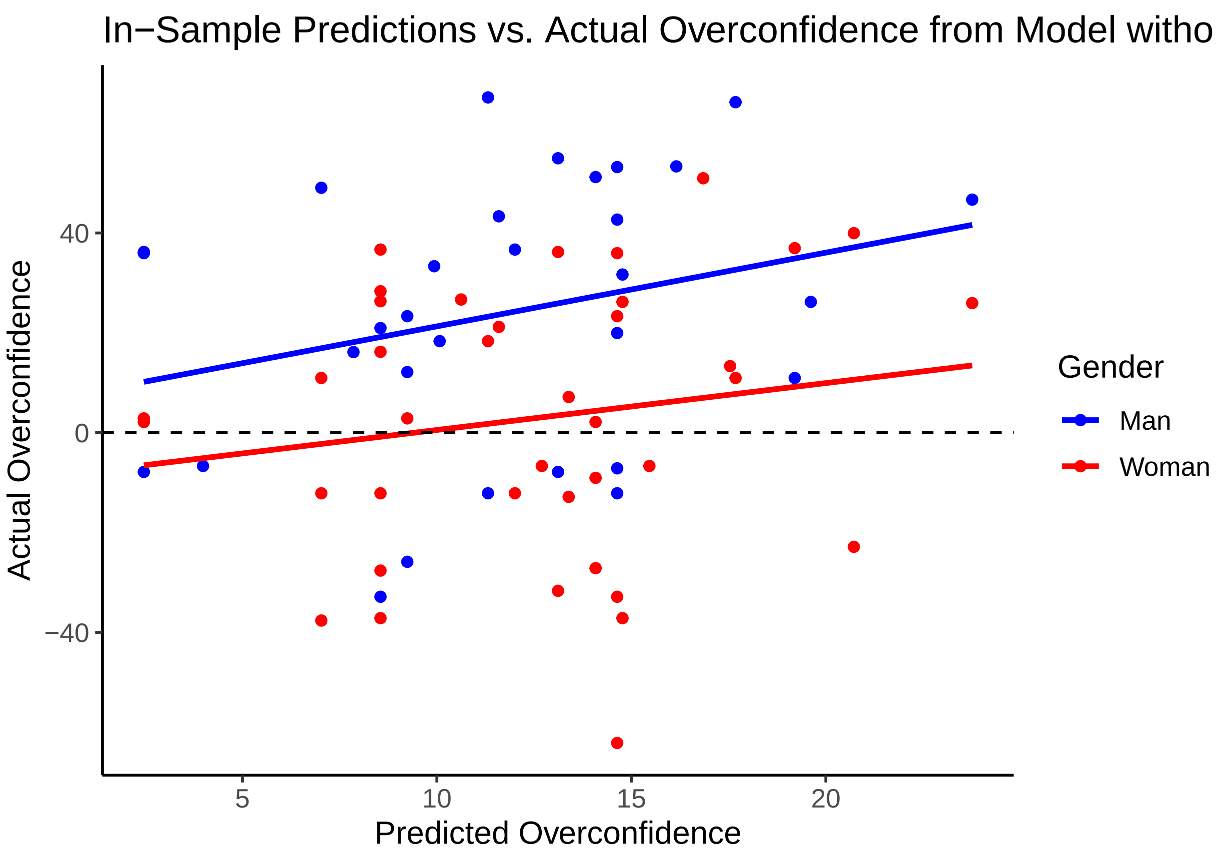





```
attention_gender %>%
  mutate(predictions = predict(lin_int_mod)) %>%
```

```r
ggplot(aes(x = predictions, y = overconfidence, color = gender)) +
geom_point() +
geom_smooth(method = 'lm', se = FALSE) +
geom_hline(yintercept = 0, linetype = 'dashed') +
theme_classic() +
labs(x = "Predicted Overconfidence", y = 'Actual Overconfidence', color = "Gender") +
scale_color_manual(values = c("M" = 'blue', "W" = 'red'),
                   labels = c("M" = "Man", "W" = "Woman")) +
ggtitle('In-Sample Predictions vs. Actual Overconfidence from Model without Gender') +
theme(text=element_text(size=12))
```

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



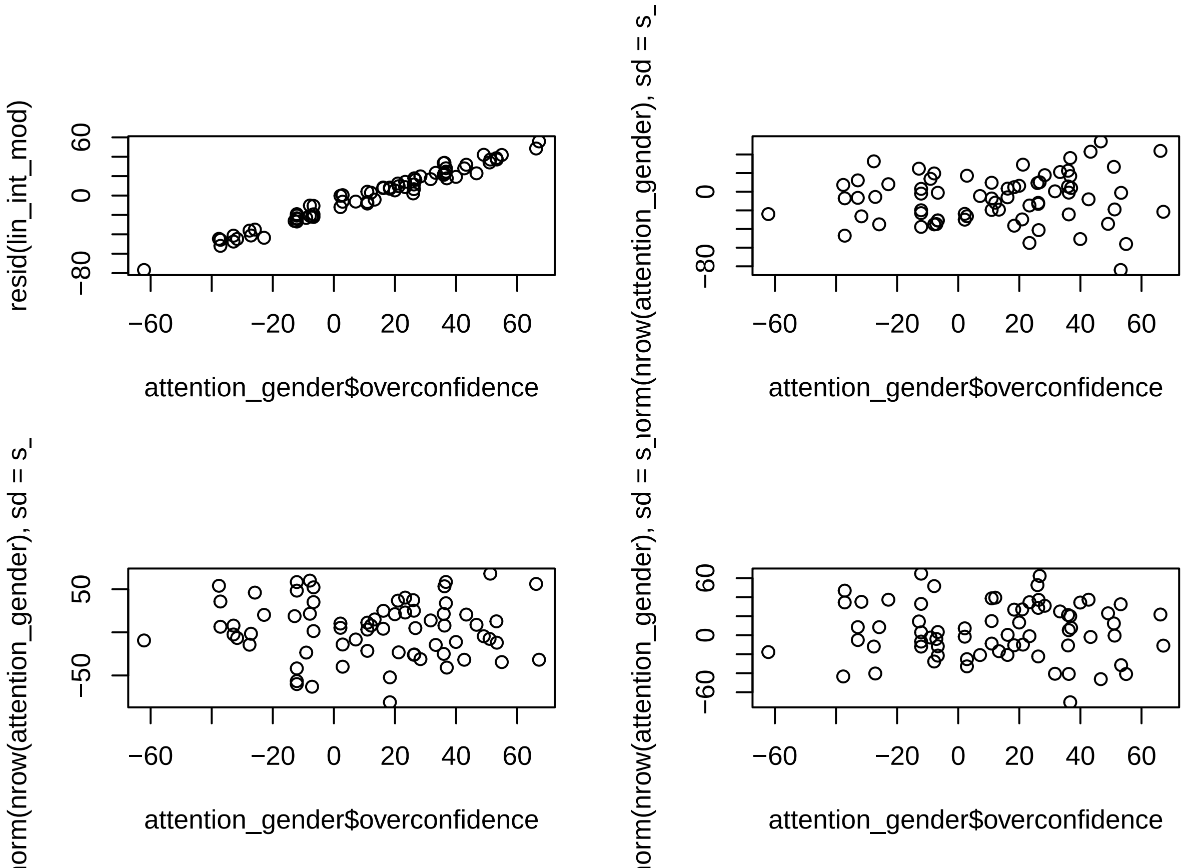

```r
# Residual versus fitted shows constant variance

# Q-Q plot shows normality

# Actual y versus predicted y shows us if something might be missed by not expressing gender
```

Checking for Overconfidence versus residual.

```r
s_hat = sd(resid(lin_int_mod))
par(mfrow = c(2, 2))
plot(attention_gender$overconfidence, resid(lin_int_mod))
replicate(3,
          plot(attention_gender$overconfidence, rnorm(nrow(attention_gender), sd = s_hat)))
```




```
## [[1]]
## NULL
##
## [[2]]
## NULL
##
## [[3]]
## NULL
```

```
par(mfrow = c(1, 1))
# this shows us that our model does VERY bad for the extremes and predicts small values only (hovering
# aligns with the intuition that none of the coefficients are significant, so basically there's no diff
# nothing really fits this data well
```

This shows us that our model does VERY bad for the extremes and predicts small values only (hovering around 0), which aligns with the intuition that none of the coefficients are significant, so basically there's no difference from just predicting the mean of overconfidence.

Testing on the Hold Out Set

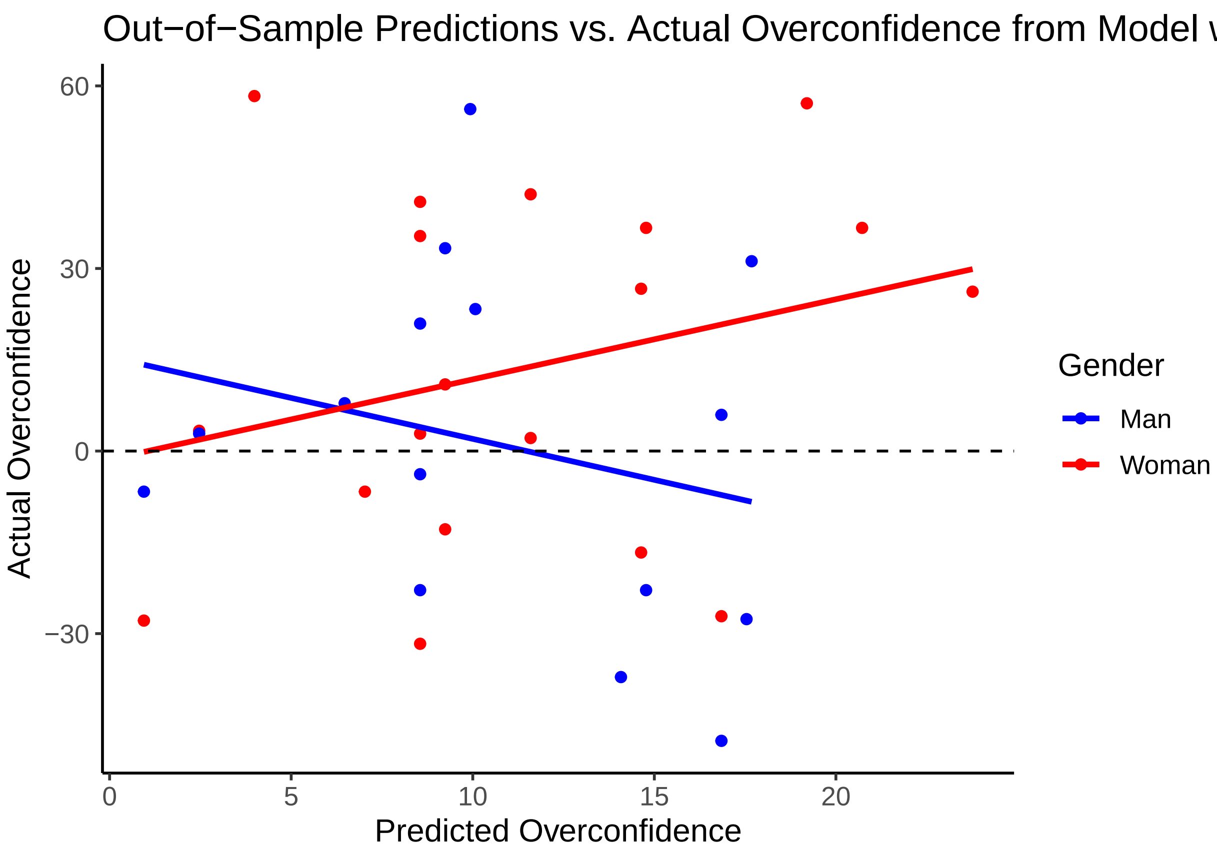
```
hold_out_set = hold_out_set %>%
  mutate(overconfidence = EstPerc - ActPerc)

preds_hold_out = predict(lin_int_mod, hold_out_set, type = 'response')

resids_hold_out = preds_hold_out - hold_out_set$overconfidence
```

```
hold_out_set %>%
  mutate(predictions = predict(lin_int_mod, hold_out_set)) %>%
  ggplot(aes(x = predictions, y = overconfidence, color = gender)) +
  geom_point() +
  geom_smooth(method = 'lm', se = FALSE) +
  geom_hline(yintercept = 0, linetype = 'dashed') +
  theme_classic() +
  labs(x = "Predicted Overconfidence", y = 'Actual Overconfidence', color = "Gender") +
  scale_color_manual(values = c("M" = 'blue', "W" = 'red'),
                     labels = c("M" = "Man", "W" = "Woman")) +
  ggtitle('Out-of-Sample Predictions vs. Actual Overconfidence from Model without Gender') +
  theme(text=element_text(size=12))
```

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



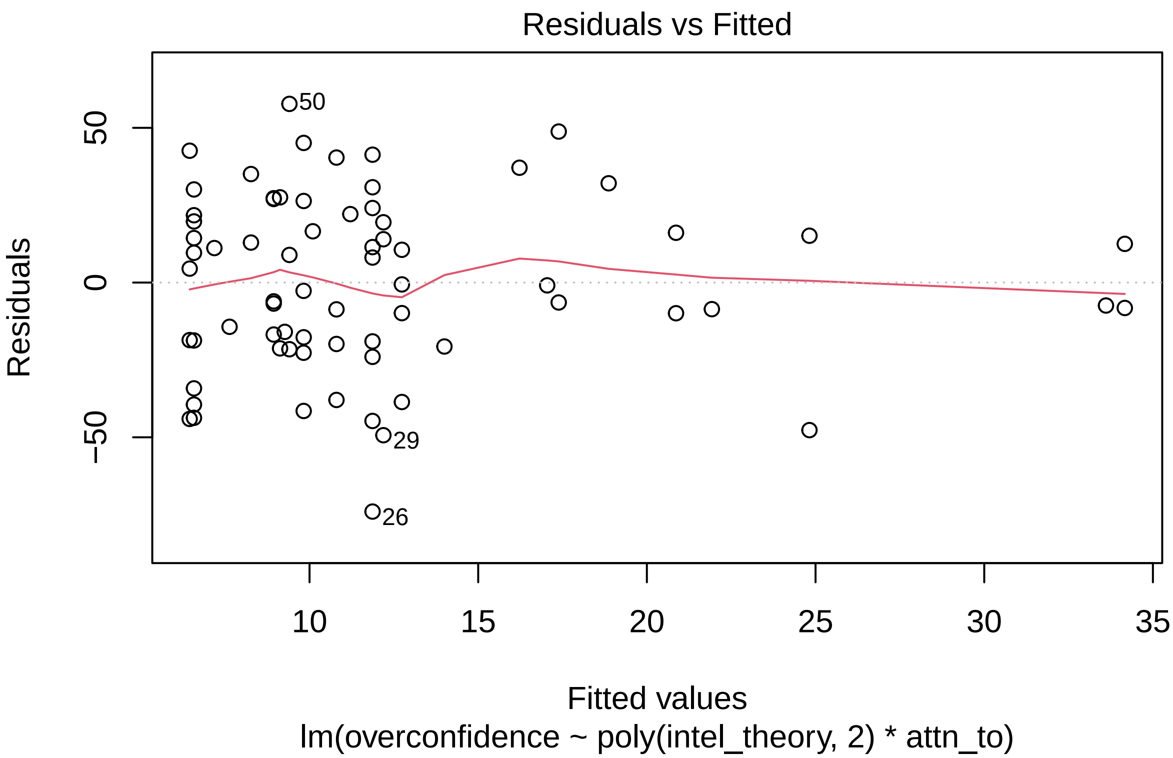
Attempting non-linear models, violation of assumptions ensue.

```
quad_int_mod = lm(overconfidence ~ poly(intel_theory,2) * attn_to, data = attention_gender)
summary(quad_int_mod)
```

```
##
## Call:
## lm(formula = overconfidence ~ poly(intel_theory, 2) * attn_to,
##     data = attention_gender)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -74.008 -18.935  -0.752  21.256  57.736
##
## Coefficients:
##                                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)                              12.831      5.683   2.258   0.0274 *
## poly(intel_theory, 2)1                   26.862     45.028   0.597   0.5529
## poly(intel_theory, 2)2                   31.767     42.193   0.753   0.4543
## attn_tohardprobs                         -1.218      7.243  -0.168   0.8669
## poly(intel_theory, 2)1:attn_tohardprobs  18.471     59.732   0.309   0.7581
## poly(intel_theory, 2)2:attn_tohardprobs   4.684     59.217   0.079   0.9372
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 29.24 on 64 degrees of freedom
## Multiple R-squared:  0.04829,    Adjusted R-squared:  -0.02606
## F-statistic: 0.6495 on 5 and 64 DF,  p-value: 0.6629
```

```
plot(quad_int_mod, which=1)
```

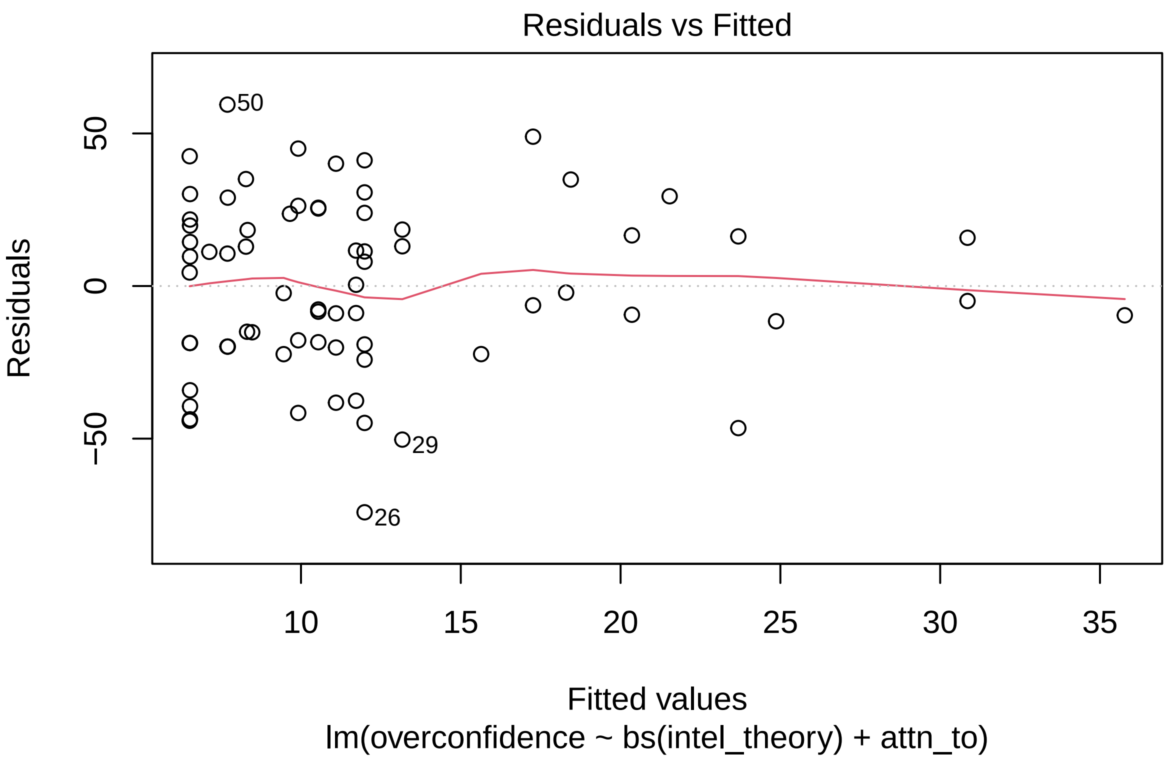



```
bs_spline_mod = lm(overconfidence ~ bs(intel_theory) + attn_to, data = attention_gender)
summary(bs_spline_mod)
```

```
##
## Call:
## lm(formula = overconfidence ~ bs(intel_theory) + attn_to, data = attention_gender)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -74.13 -19.02  -0.87  21.30  59.44
##
## Coefficients:
```

```
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)           18.297     18.273   1.001    0.320
## bs(intel_theory)1  -23.510     48.313  -0.487    0.628
## bs(intel_theory)2   -3.857     26.686  -0.145    0.886
## bs(intel_theory)3   17.480     29.896   0.585    0.561
## attn_tohardprobs    -1.181      7.199  -0.164    0.870
##
## Residual standard error: 29.02 on 65 degrees of freedom
## Multiple R-squared:  0.04733,    Adjusted R-squared:  -0.01129
## F-statistic: 0.8074 on 4 and 65 DF,  p-value: 0.525
```

```
plot(bs_spline_mod, which=1)
```



From this analysis, it's not worth using the non-linear models because they have a clear pattern in their residuals and this would just violate the assumptions of the lm() model grossly. Therefore, we will stick to linear models with interaction terms.

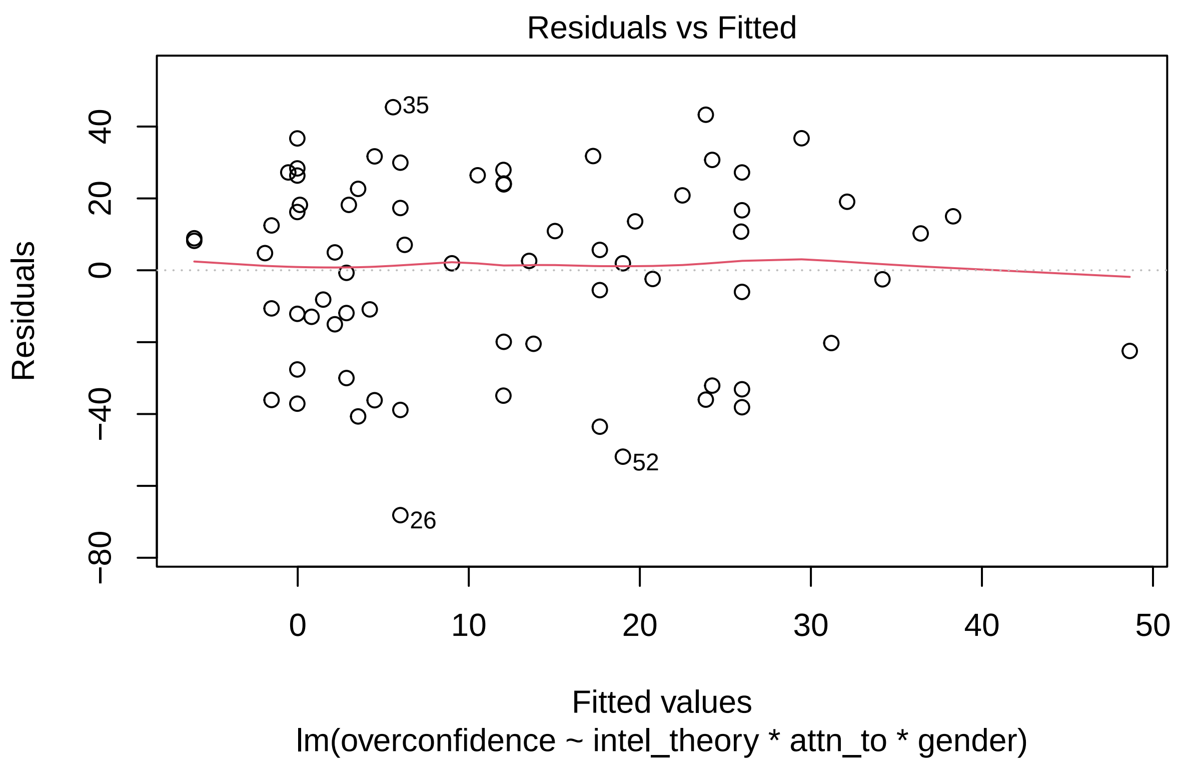
Model with Gender, with interaction terms.

```
lin_mod_gen_int = lm(overconfidence ~ intel_theory * attn_to * gender, data = attention_gender)
summary(lin_mod_gen_int)
```

```
##
## Call:
## lm(formula = overconfidence ~ intel_theory * attn_to * gender,
##     data = attention_gender)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -68.141 -20.160   4.873  20.397  45.378
##
```

```
## Coefficients:
##                                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)                             1.137     21.521   0.053    0.958
## intel_theory                            8.262      6.620   1.248    0.217
## attn_tohardprobs                       -3.021     32.098  -0.094    0.925
## genderW                                -8.496     39.825  -0.213    0.832
## intel_theory:attn_tohardprobs          -1.298      9.355  -0.139    0.890
## intel_theory:genderW                   -5.539     11.284  -0.491    0.625
## attn_tohardprobs:genderW               -7.720     51.560  -0.150    0.881
## intel_theory:attn_tohardprobs:genderW   4.600     14.450   0.318    0.751
##
## Residual standard error: 27.54 on 62 degrees of freedom
## Multiple R-squared:  0.1817, Adjusted R-squared:  0.08936
## F-statistic: 1.967 on 7 and 62 DF,  p-value: 0.07399
```

```
plot(lin_mod_gen_int, which=1)
```

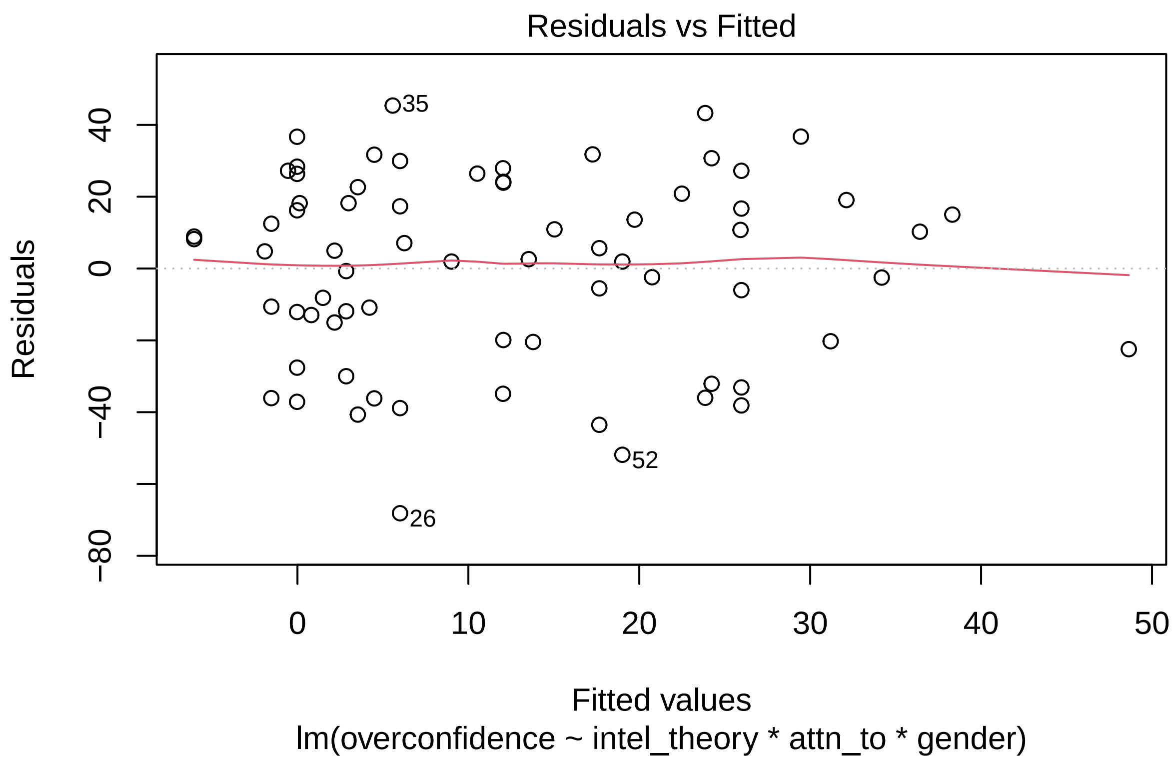


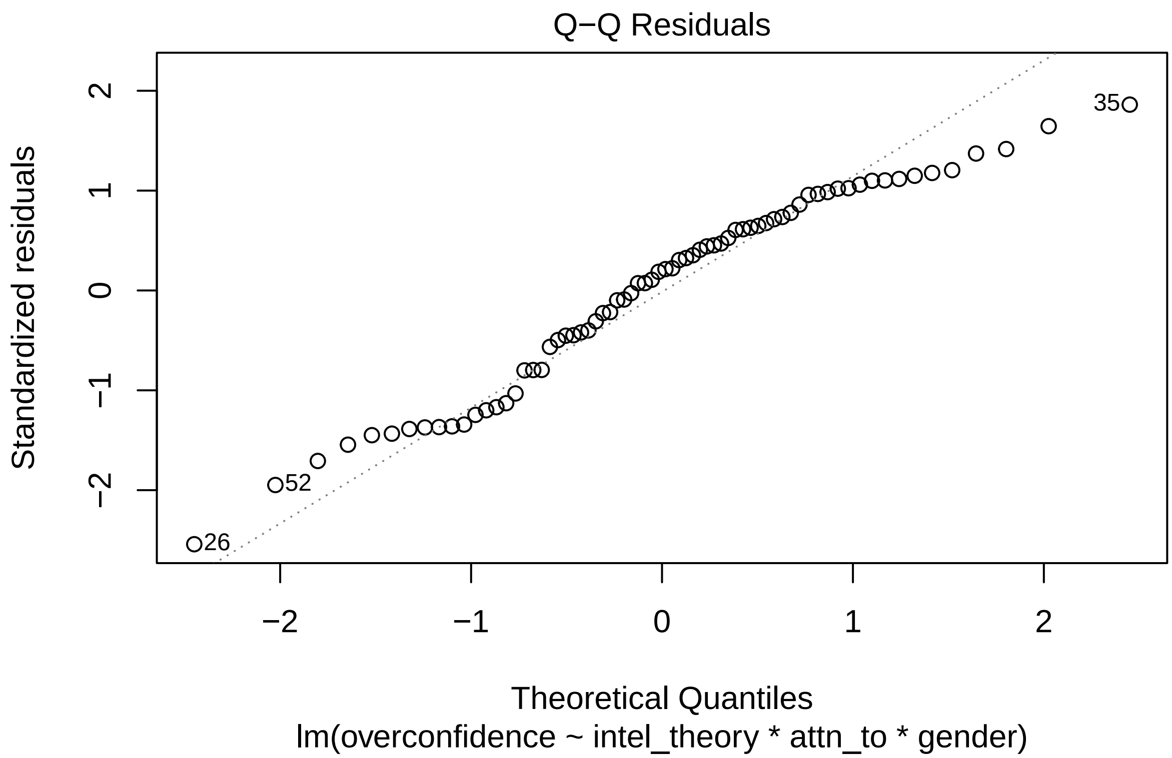

```
summary(lin_mod_gen_int)
```

```
##
## Call:
## lm(formula = overconfidence ~ intel_theory * attn_to * gender,
##     data = attention_gender)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -68.141 -20.160   4.873  20.397  45.378
##
## Coefficients:
##                                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)                             1.137     21.521   0.053    0.958
```

```
## intel_theory                               8.262      6.620   1.248     0.217
## attn_tohardprobs                          -3.021     32.098  -0.094     0.925
## genderW                                   -8.496     39.825  -0.213     0.832
## intel_theory:attn_tohardprobs             -1.298      9.355  -0.139     0.890
## intel_theory:genderW                      -5.539     11.284  -0.491     0.625
## attn_tohardprobs:genderW                  -7.720     51.560  -0.150     0.881
## intel_theory:attn_tohardprobs:genderW      4.600     14.450   0.318     0.751
##
## Residual standard error: 27.54 on 62 degrees of freedom
## Multiple R-squared:  0.1817, Adjusted R-squared:  0.08936
## F-statistic: 1.967 on 7 and 62 DF,  p-value: 0.07399
```

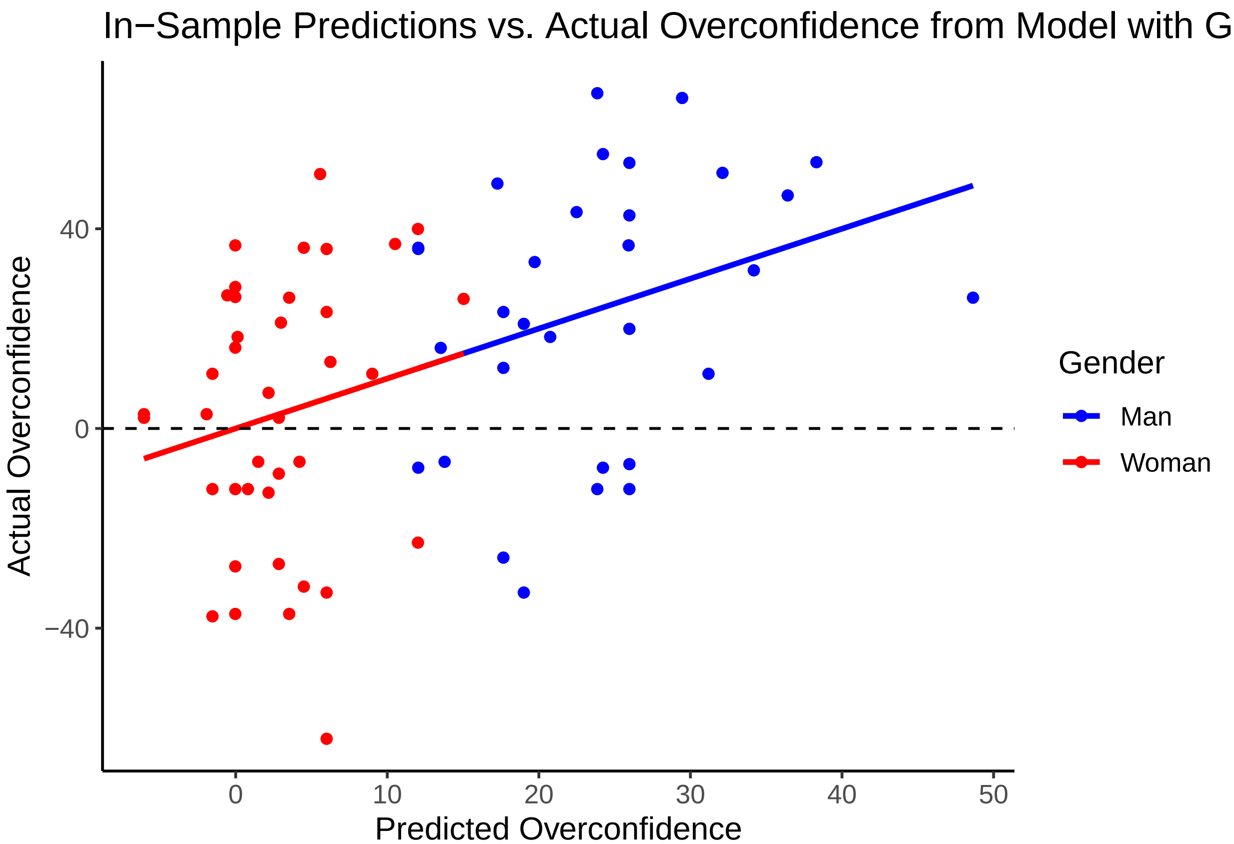
```r
for(i in 1:2){
  plot(lin_mod_gen_int, which=i)
}
```



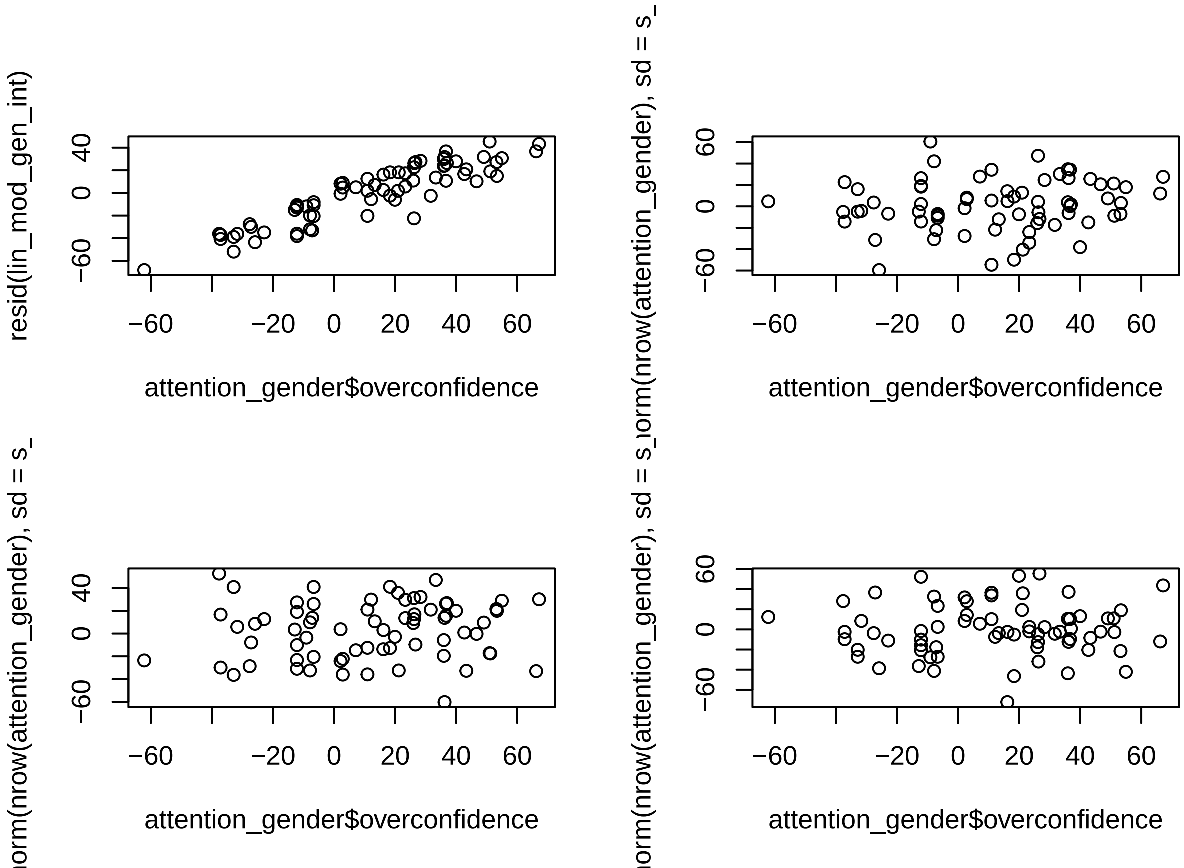



```r
attention_gender %>%
  mutate(predictions = predict(lin_mod_gen_int)) %>%
  ggplot(aes(x = predictions, y = overconfidence, color = gender)) +
  geom_point() +
  geom_smooth(method = 'lm', se = FALSE) +
  geom_hline(yintercept = 0, linetype = 'dashed') +
  theme_classic() +
  labs(x = "Predicted Overconfidence", y = 'Actual Overconfidence', color = "Gender") +
  scale_color_manual(values = c("M" = 'blue', "W" = 'red'),
                     labels = c("M" = "Man", "W" = "Woman")) +
  ggtitle('In-Sample Predictions vs. Actual Overconfidence from Model with Gender') +
  theme(text=element_text(size=12))
```

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




```
s_hat = sd(resid(lin_mod_gen_int))
par(mfrow = c(2, 2))
plot(attention_gender$overconfidence, resid(lin_mod_gen_int))
replicate(3,
          plot(attention_gender$overconfidence, rnorm(nrow(attention_gender), sd = s_hat)))
```




```
## [[1]]
## NULL
##
## [[2]]
## NULL
##
## [[3]]
## NULL
```

```
par(mfrow = c(1, 1))
```

Testing on Hold Out Set

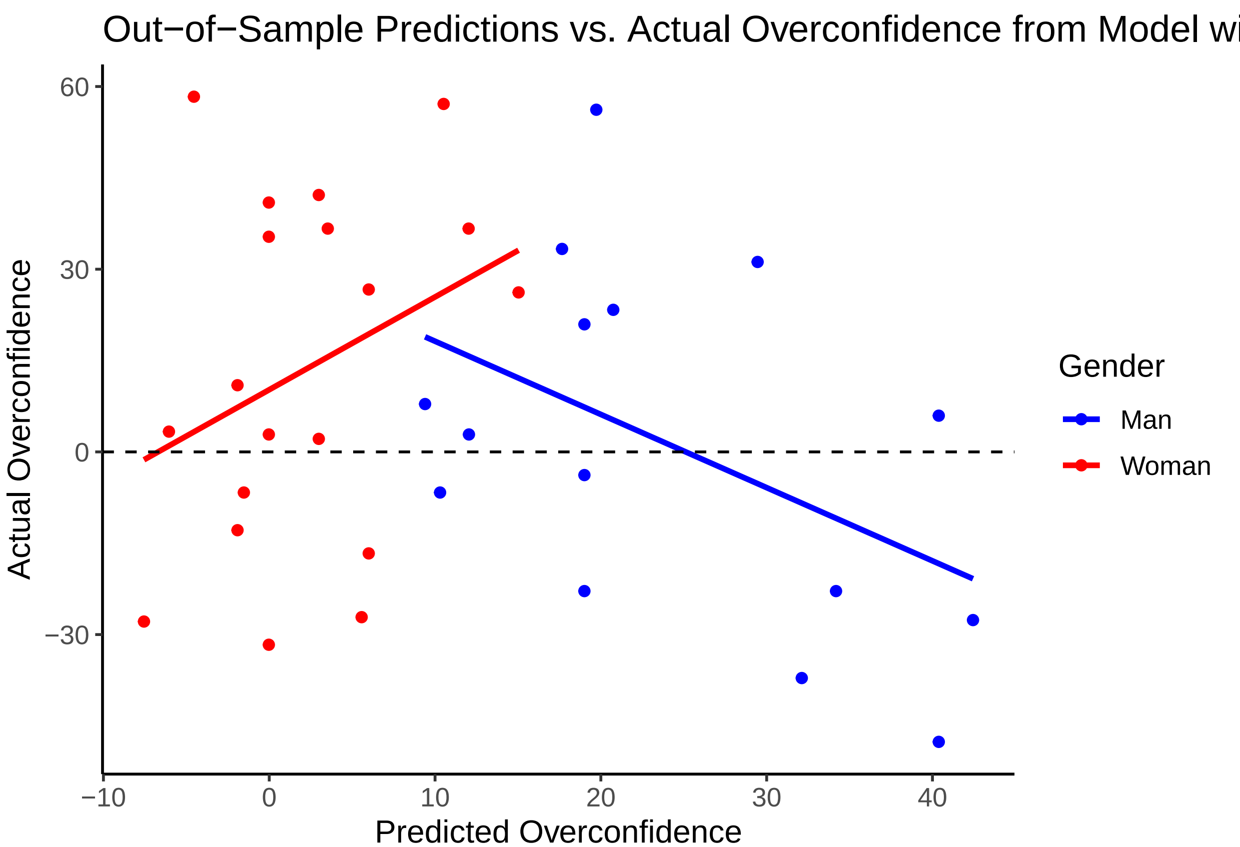
```
preds_hold_out_gen = predict(lin_mod_gen_int, hold_out_set, type = 'response')

resids_hold_out_gen = preds_hold_out_gen - hold_out_set$overconfidence
```

```
hold_out_set %>%
  mutate(predictions = predict(lin_mod_gen_int, hold_out_set)) %>%
  ggplot(aes(x = predictions, y = overconfidence, color = gender)) +
  geom_point() +
  geom_smooth(method = 'lm', se = FALSE) +
  geom_hline(yintercept = 0, linetype = 'dashed') +
  theme_classic() +
  labs(x = "Predicted Overconfidence", y = 'Actual Overconfidence', color = "Gender") +
  scale_color_manual(values = c("M" = 'blue', "W" = 'red'),
```

```
                     labels = c("M" = "Man", "W" = "Woman")) +
  ggtitle('Out-of-Sample Predictions vs. Actual Overconfidence from Model with Gender') +
  theme(text=element_text(size=12))
```

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



In-sample Mean-Squared Error (MSE)

```
mse_nogender = mean(residuals(lin_int_mod)^2)

mse_gender = mean(residuals(lin_mod_gen_int)^2)

print(mse_nogender); print(mse_gender)
```

```
## [1] 797.7474
## [1] 671.848
```

The MSE for the model with no-gender is higher than the one with gender.

Percent decrease of MSE with the addition of gender.

```
(mse_nogender - mse_gender) / ((mse_nogender + mse_gender) / 2)
```

```
## [1] 0.1713388
```

Out-of-Sample MSE

```
non_gen_hold_out_mse = mean(resids_hold_out^2)
gen_hold_out_mse = mean(resids_hold_out_gen^2)

non_gen_hold_out_mse; gen_hold_out_mse
```

```
## [1] 850.8773
```

```
## [1] 1246.537
```

When tested out of sample, the model with gender performs worse in MSE.

ANOVA test

Same assumptions to preform linear model so we can perform this.

```
anova(lin_int_mod, lin_mod_gen_int)
```

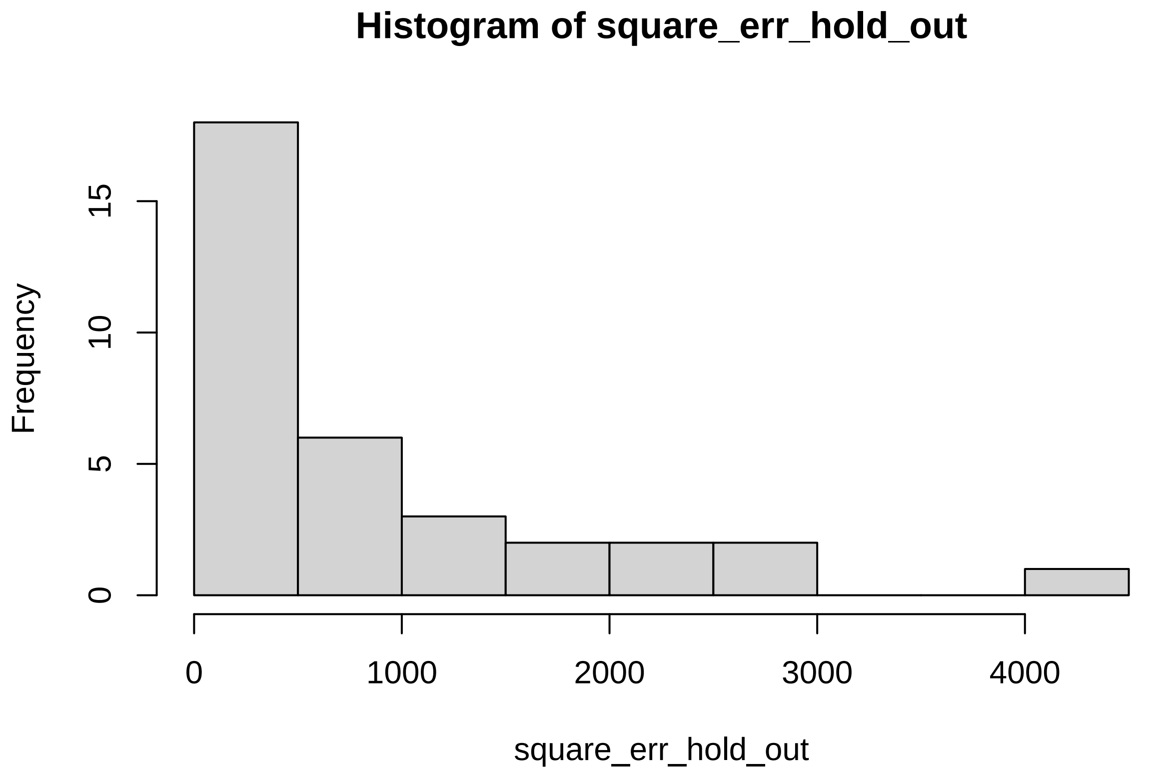
```
## Analysis of Variance Table
##
## Model 1: overconfidence ~ intel_theory * attn_to
## Model 2: overconfidence ~ intel_theory * attn_to * gender
##   Res.Df   RSS Df Sum of Sq      F Pr(>F)
## 1     66 55842
## 2     62 47029  4      8813 2.9046 0.0287 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Adding gender does make a signficant difference. Small sample set with noisy data, so would need to have more data to possibly come up with something. # Wilcoxon Signed Ranks Test

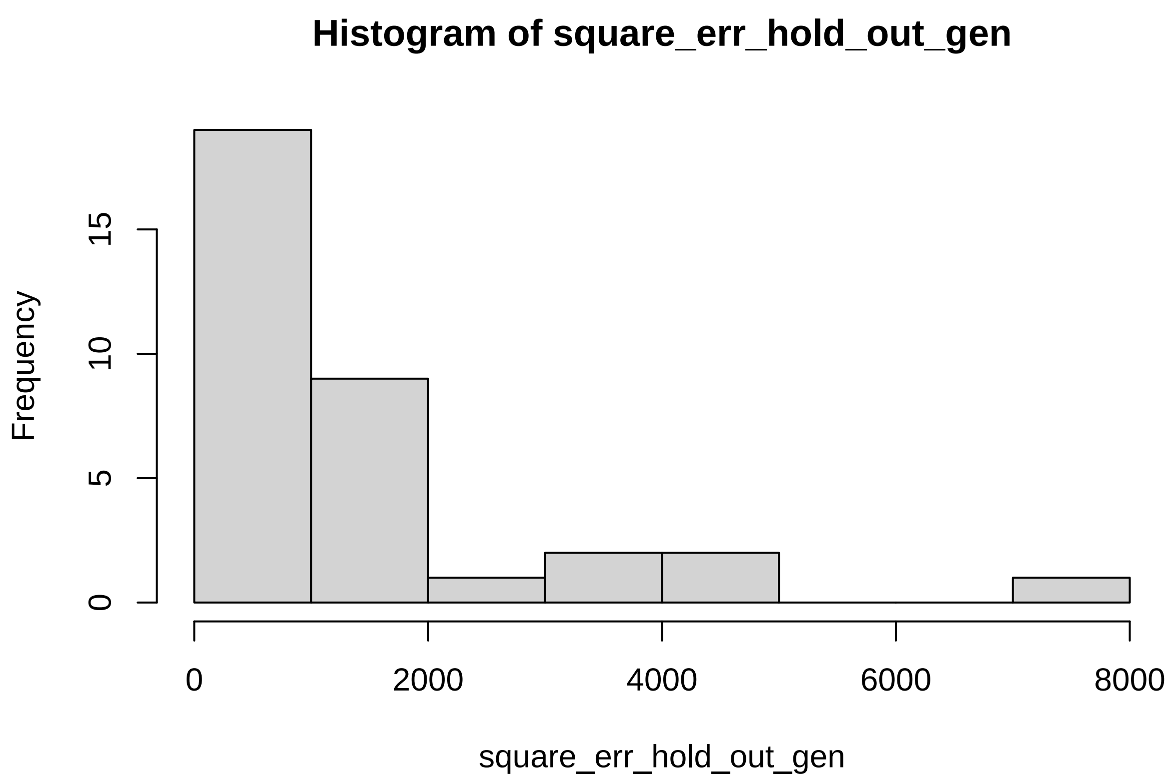
Checking the loose assumption of symmetry of squared residuals.

```
square_err_hold_out = resids_hold_out^2
square_err_hold_out_gen = resids_hold_out_gen^2

hist(square_err_hold_out)
```



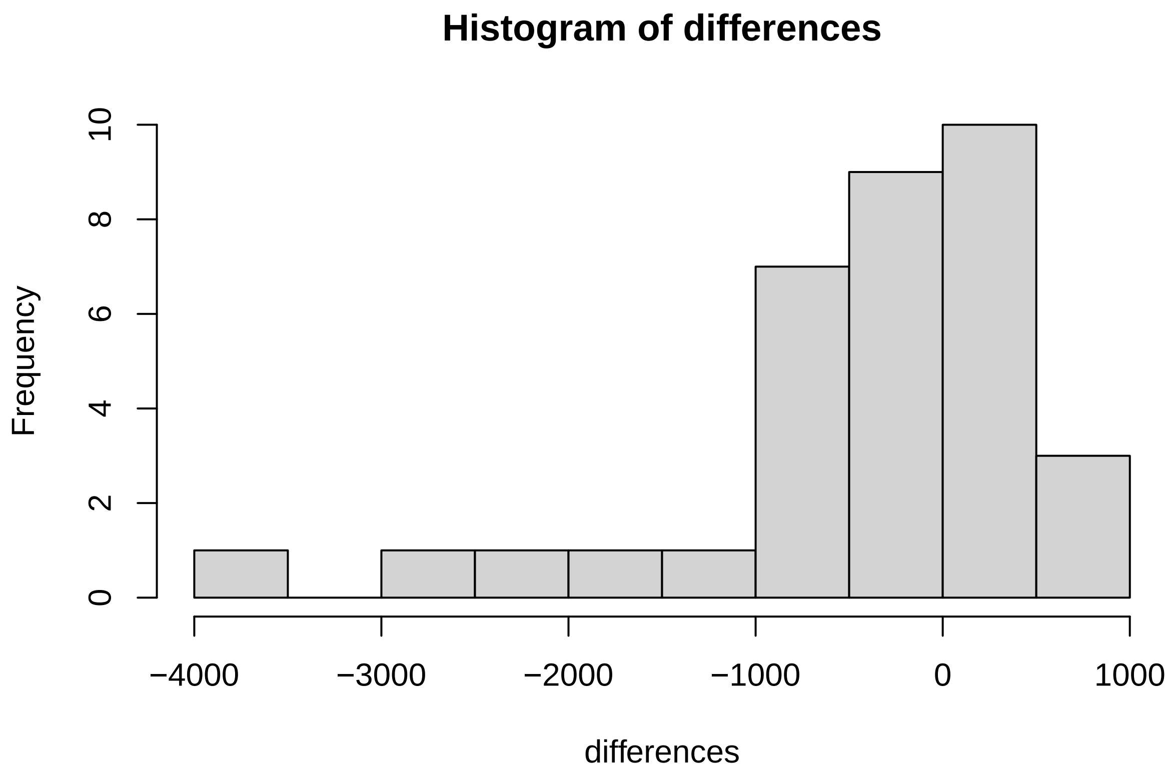

```
hist(square_err_hold_out_gen)
```



These distributions are not entirely different, however the squared errors for the model which include gender is much more right skewed. There is one squared error which is around 7000 to 8000, which is probably why the MSE for the model with gender is higher. But, this is only driven by a single point. The Wilcoxon test does have a flexibility assumption of symmetry in the models but if violated it just makes the results less strong, but doesn't invalidate the test entirely.

Checking the distribution of the differences of squared errors.

```
differences = square_err_hold_out - square_err_hold_out_gen
hist(differences)
```




```
median(differences)
```

```
## [1] -140.5247
```

On average, the differences between the squared error between the two models on the held out data set hovers around 0, except one point which has a different of -4000. This is again distorting the results, making it seem like the model without gender is superior which might not actually be the case. The data may just be more sparse and that point could be seen as an outlier.

```
wilcox_test_result <- wilcox.test(square_err_hold_out, square_err_hold_out_gen, paired = TRUE, alternati

wilcox_test_result
```

```
##
##  Wilcoxon signed rank exact test
##
## data:  square_err_hold_out and square_err_hold_out_gen
## V = 188, p-value = 0.06182
## alternative hypothesis: true location shift is not equal to 0
```

Using the Wilcoxon Signed Ranks Test to see if these models perform significantly different on the held-out data, using an α level of 0.05, these two models do not perform significantly different on the held out data. However, it's important to note that the conclusion changes when using an α level of 0.1. The addition of gender helped performance on the in sample loss, but that is generally regarded as less important compared to the out-of-sample loss, which indicates that the addition of gender may have just allowed over fitting for predictions. However, more data should be collected because the p-value of the Wilcoxon signed rank test is not significant for all commonly used thresholds. In addition, since the Wilcoxon signed ranks test doesn't follow the assumption of symmetry, this makes the results more weak, which means the actual p-value of the differences between the residuals of these two models might be higher in practice with more rigorous methods.