

Data Appendix: Prompt Engineering in Generative Pre-trained Transformers (GPT) Models for Healthcare Simulation Case Scenarios

Sara Maaz et al

2024-08-10

Contents

1	Load, prepare, and explore data	1
2	Explore data	3
2.1	Example exploration with Element 1	3
2.2	Explore SSET total	3
2.3	Explore raters	4
3	Compare case quality by language model	5
3.1	Descriptive statistics and charts	5
3.2	Regression models	6
3.2.1	Null model	6
3.2.2	Main model	7
3.2.3	Interaction model (language model x prompt chaining)	12
3.2.4	Interaction model (prompt chaining x rater)	21
3.2.5	Interaction model (language model x rater)	22
4	Search for outliers which consistently scored poorly	24
5	Compare the quality of individual SSET elements across the LLM's	25

1 Load, prepare, and explore data

```
library(lme4)
```

```
## Loading required package: Matrix
```

```
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(readxl)
```

```
d <- read_excel("SSET Responses Final Identified Corrected.xlsx", sheet="Identified & Categorized")
```

```
## New names:  
## * 'Case Code' -> 'Case Code...1'  
## * 'Element 1: Learning Objectives' -> 'Element 1: Learning Objectives...5'  
## * 'Element 1: Learning Objectives' -> 'Element 1: Learning Objectives...6'  
## * 'Element 1: Learning Objectives' -> 'Element 1: Learning Objectives...7'  
## * 'Element 1: Learning Objectives' -> 'Element 1: Learning Objectives...8'  
## * 'Element 1: Learning Objectives' -> 'Element 1: Learning Objectives...9'  
## * 'Element 1: Learning Objectives' -> 'Element 1: Learning Objectives...10'  
## * 'Element 1: Learning Objectives' -> 'Element 1: Learning Objectives...11'  
## * 'Element II : Clinical Context/ Scenario Overview' -> 'Element II : Clinical  
## Context/ Scenario Overview...13'  
## * 'Element II : Clinical Context/ Scenario Overview' -> 'Element II : Clinical  
## Context/ Scenario Overview...14'  
## * 'Element III: Critical Actions' -> 'Element III: Critical Actions...16'  
## * 'Element III: Critical Actions' -> 'Element III: Critical Actions...17'  
## * 'Element III: Critical Actions' -> 'Element III: Critical Actions...18'  
## * 'Element IV: Patient States' -> 'Element IV: Patient States...20'  
## * 'Element IV: Patient States' -> 'Element IV: Patient States...21'  
## * 'Element IV: Patient States' -> 'Element IV: Patient States...22'  
## * 'Element IV: Patient States' -> 'Element IV: Patient States...23'  
## * 'Element V: Scenario, Materials and Resources' -> 'Element V: Scenario,  
## Materials and Resources...25'  
## * 'Element V: Scenario, Materials and Resources' -> 'Element V: Scenario,  
## Materials and Resources...26'  
## * 'Element VI: Debriefing Plan' -> 'Element VI: Debriefing Plan...28'  
## * 'Element VI: Debriefing Plan' -> 'Element VI: Debriefing Plan...29'  
## * 'Case Code' -> 'Case Code...35'
```

```
d$SSETtotal <- d$`E1 Total` + d$`E 2 Total` + d$`E3 Total` + d$`E4 Total` + d$`E5 Total` + d$`E6 total`
```

2 Explore data

2.1 Example exploration with Element 1

```
# RESULTS NOT SHOWN
```

```
summary(d$`Element 1: Learning Objectives...5`)  
# hist(d$`Element 1: Learning Objectives...5`)
```

```
with(d, table(`Element 1: Learning Objectives...5`, `Is this a prompt chaining case?`, useNA = 'always'))
```

```
with(d, round(prop.table(table(`Element 1: Learning Objectives...5`, `Is this a prompt chaining case?`, useNA = 'always'))))
```

```
with(d, round(prop.table(table(`Element 1: Learning Objectives...5`, `Is this a prompt chaining case?`, useNA = 'always'))))
```

```
boxplot(`Element 1: Learning Objectives...5` ~ `Is this a prompt chaining case?`, d)
```

```
# RESULTS NOT SHOWN
```

```
summary(reg1 <- lm(`Element 1: Learning Objectives...5` ~ `Is this a prompt chaining case?`, d))
```

```
# RESULTS NOT SHOWN
```

```
if (!require(dplyr)) install.packages('dplyr')  
library(dplyr)
```

```
dplyr::group_by(d, `Is this a prompt chaining case?`) %>%  
dplyr::summarise(  
  count = n(),  
  mean = mean(`Element 1: Learning Objectives...5`, na.rm = TRUE),  
  sd = sd(`Element 1: Learning Objectives...5`, na.rm = TRUE)  
)
```

```
2.92-2.54
```

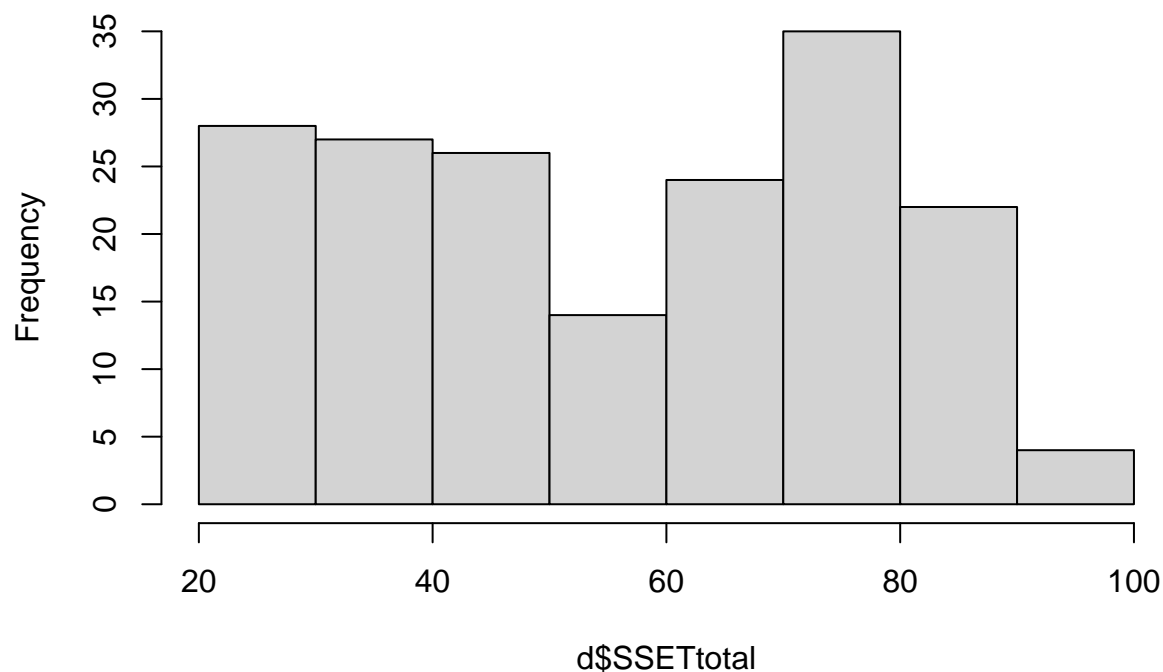
2.2 Explore SSET total

```
summary(d$SSETtotal)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   
##    20.00  37.75   56.00   56.17  75.00   93.00
```

```
hist(d$SSETtotal)
```

Histogram of d\$SSETtotal



2.3 Explore raters

```
with(d, addmargins(table(`Language model`,Rater)))
```

```
##           Rater
## Language model  1  2  3 Sum
##      Bard      30 15 15 60
##    Bing Prec.  30 13 17 60
##      GPT-4     30 17 13 60
##      Sum      90 45 45 180
```

```
with(d, addmargins(table(`Is this a prompt chaining case?`,Rater, useNA = 'always')))
```

```
##           Rater
## Is this a prompt chaining case?  1  2  3 <NA> Sum
##                               No  61 30 30  0 121
##                               Yes  29 15 15  0 59
##                               <NA>  0  0  0  0  0
##                               Sum  90 45 45  0 180
```

```
# RESULTS NOT SHOWN
```

```
View(d %>% filter(is.na(`Is this a prompt chaining case?`)))
```

C28 and S58 are missing prompt chaining/zero shot information, on Sep 18 2023

```
with(d, addmargins(table(`Is this a prompt chaining case?`,Rater,`Language model`)))
```

```
## , , Language model = Bard
##
##               Rater
## Is this a prompt chaining case?  1  2  3 Sum
##                               No  20 10 10 40
##                               Yes 10  5  5 20
##                               Sum 30 15 15 60
##
## , , Language model = Bing Prec.
##
##               Rater
## Is this a prompt chaining case?  1  2  3 Sum
##                               No  21 10 10 41
##                               Yes  9  3  7 19
##                               Sum 30 13 17 60
##
## , , Language model = GPT-4
##
##               Rater
## Is this a prompt chaining case?  1  2  3 Sum
##                               No  20 10 10 40
##                               Yes 10  7  3 20
##                               Sum 30 17 13 60
##
## , , Language model = Sum
##
##               Rater
## Is this a prompt chaining case?  1  2  3 Sum
##                               No  61 30 30 121
##                               Yes 29 15 15 59
##                               Sum 90 45 45 180
```

3 Compare case quality by language model

Research question: Is there a difference in SSET result when comparing cases generated by the three different language models?

3.1 Descriptive statistics and charts

```
dplyr::group_by(d, `Language model`) %>%
dplyr::summarise(
  count = n(),
  mean = mean(SSETtotal, na.rm = TRUE),
```

```
sd = sd(SSETtotal, na.rm = TRUE)
)
```

```
## # A tibble: 3 x 4
##   'Language model' count  mean    sd
##   <chr>           <int> <dbl> <dbl>
## 1 Bard             60  51.4  16.5
## 2 Bing Prec.       60  43.4  15.2
## 3 GPT-4            60  73.8  17.9
```

3.2 Regression models

```
d$Rater.cat <- as.factor(d$Rater)
```

3.2.1 Null model

```
jtools::summ(regLM1 <- lmer(SSETtotal ~ 1 + (1|`Case Code...1`), data=d))
```

```
## Registered S3 methods overwritten by 'broom':
##   method      from
##   tidy.glht    jtools
##   tidy.summary.glht jtools

## MODEL INFO:
## Observations: 180
## Dependent Variable: SSETtotal
## Type: Mixed effects linear regression
##
## MODEL FIT:
## AIC = 1575.80, BIC = 1585.38
## Pseudo-R2 (fixed effects) = 0.00
## Pseudo-R2 (total) = 0.55
##
## FIXED EFFECTS:
## -----
##               Est.   S.E.   t val.   d.f.   p
## -----
## (Intercept)   56.17   1.94   28.94   89.00  0.00
## -----
##
## p values calculated using Kenward-Roger standard errors and d.f.
##
## RANDOM EFFECTS:
## -----
##      Group      Parameter   Std. Dev.
## -----
## Case Code...1 (Intercept)   15.48
## Residual                   14.11
```

```
## -----
##
## Grouping variables:
## -----
##      Group      # groups  ICC
## -----
## Case Code...1      90      0.55
## -----
```

3.2.2 Main model

```
library(jtools)
jtools::summ(regLM1 <- lmer(SSETtotal ~ `Language model` + Rater.cat + `Is this a prompt chaining case?`

## MODEL INFO:
## Observations: 180
## Dependent Variable: SSETtotal
## Type: Mixed effects linear regression
##
## MODEL FIT:
## AIC = 1445.28, BIC = 1470.82
## Pseudo-R2 (fixed effects) = 0.49
## Pseudo-R2 (total) = 0.77
##
## FIXED EFFECTS:
## -----
##                               Est.      2.5%   97.5%   t val.    d.f.    p
## -----
## (Intercept)                  45.72    40.36   51.07    16.73    100.78  0.00
## Language model'Bing Prec.    -7.87   -14.64   -1.11    -2.28    85.70  0.03
## Language model'GPT-4         22.18    15.42   28.95     6.43    85.70  0.00
## Rater.cat2                   16.38    12.44   20.33     8.10   106.81  0.00
## Rater.cat3                   10.83     6.88   14.77     5.36   106.63  0.00
## Is this a prompt chaining
## case?'Yes                    -3.41    -9.20    2.38    -1.15    93.08  0.25
## -----
##
## p values calculated using Kenward-Roger standard errors and d.f.
##
## RANDOM EFFECTS:
## -----
##      Group      Parameter      Std. Dev.
## -----
## Case Code...1  (Intercept)    11.28
## Residual              10.13
## -----
##
## Grouping variables:
## -----
##      Group      # groups  ICC
## -----
## Case Code...1      90      0.55
```

```
## -----
```

```
# Diagnostic tests
```

```
library(car)
```

```
## Loading required package: carData
```

```
##
```

```
## Attaching package: 'car'
```

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

```
##
```

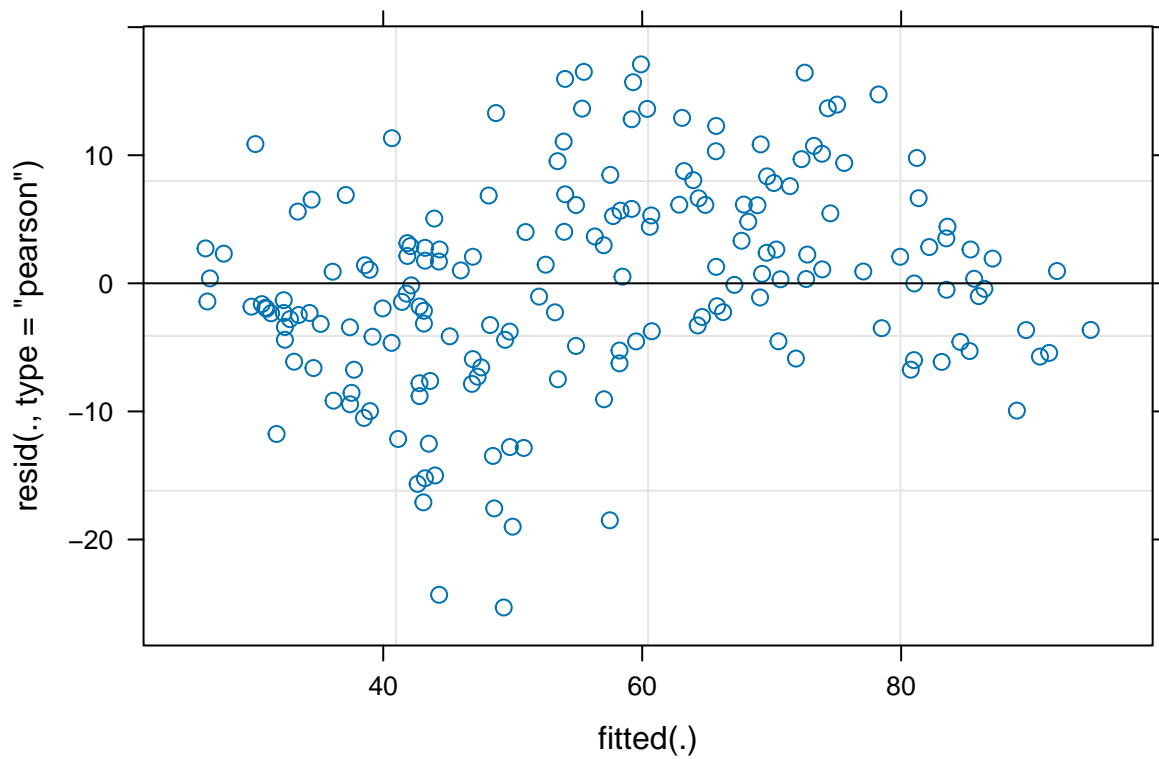
```
## recode
```

```
vif(regLM1)
```

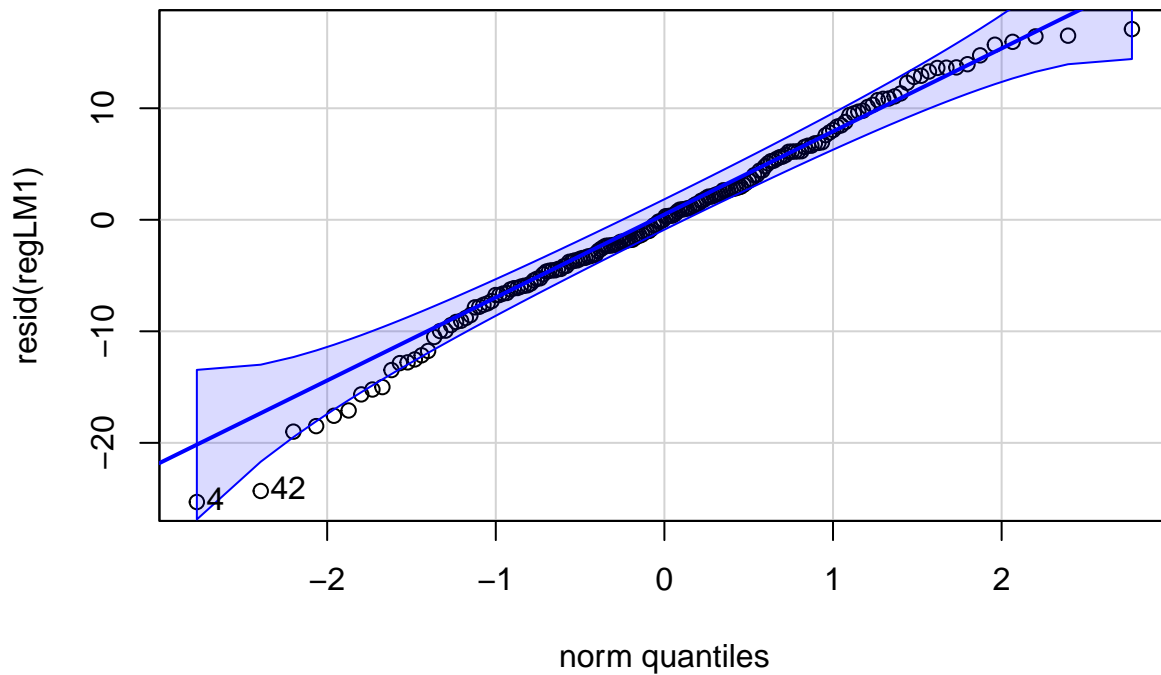
```
##
```

	GVIF	Df	GVIF^(1/(2*Df))
## 'Language model'	1.002906	2	1.000726
## Rater.cat	1.003295	2	1.000823
## 'Is this a prompt chaining case?'	1.000913	1	1.000456

```
plot(regLM1)
```




```
car::qqPlot(resid(regLM1))
```

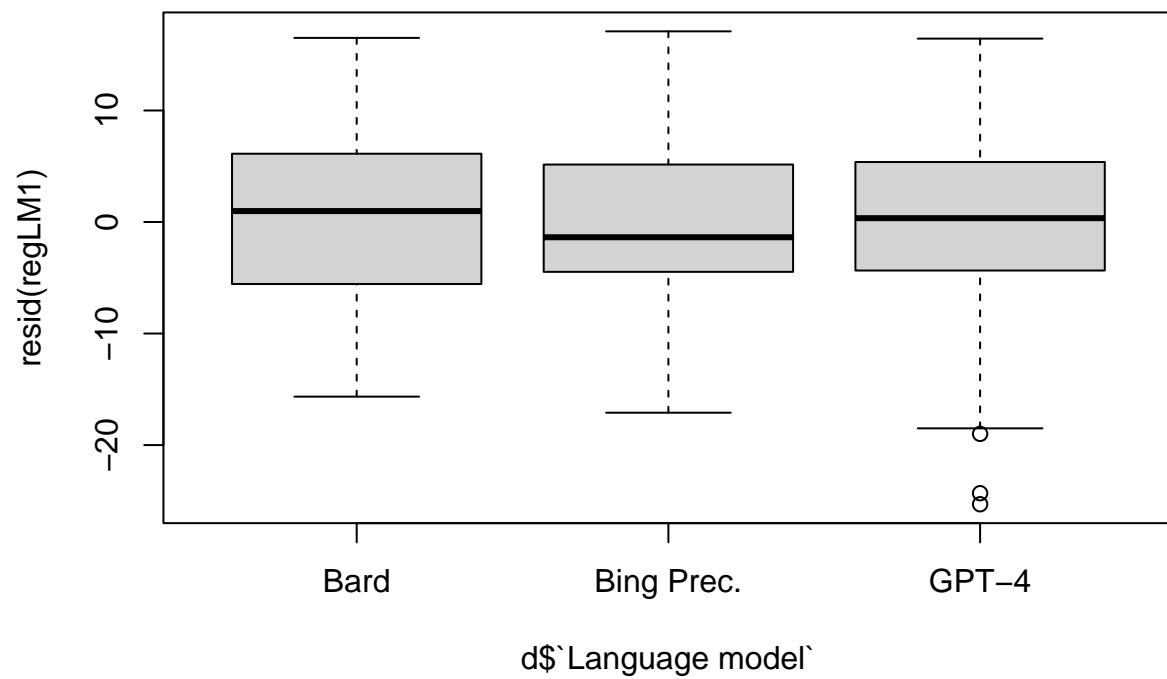


```
## [1] 4 42
```

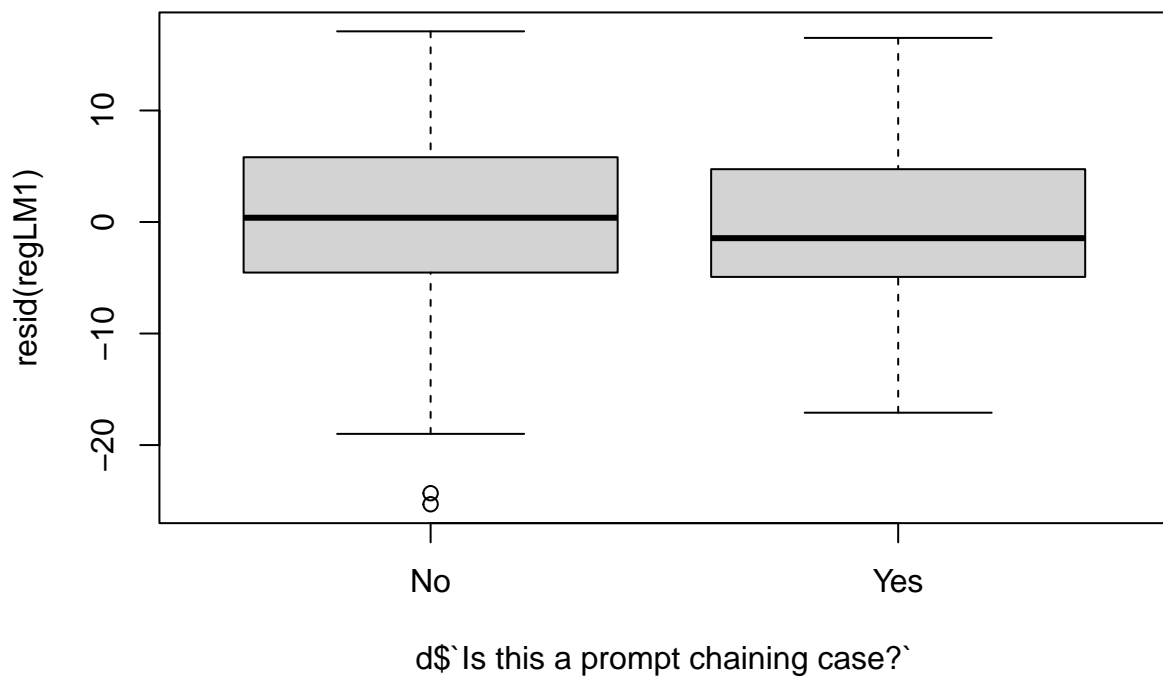
```
shapiro.test(resid(regLM1))
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: resid(regLM1)  
## W = 0.98797, p-value = 0.1293
```

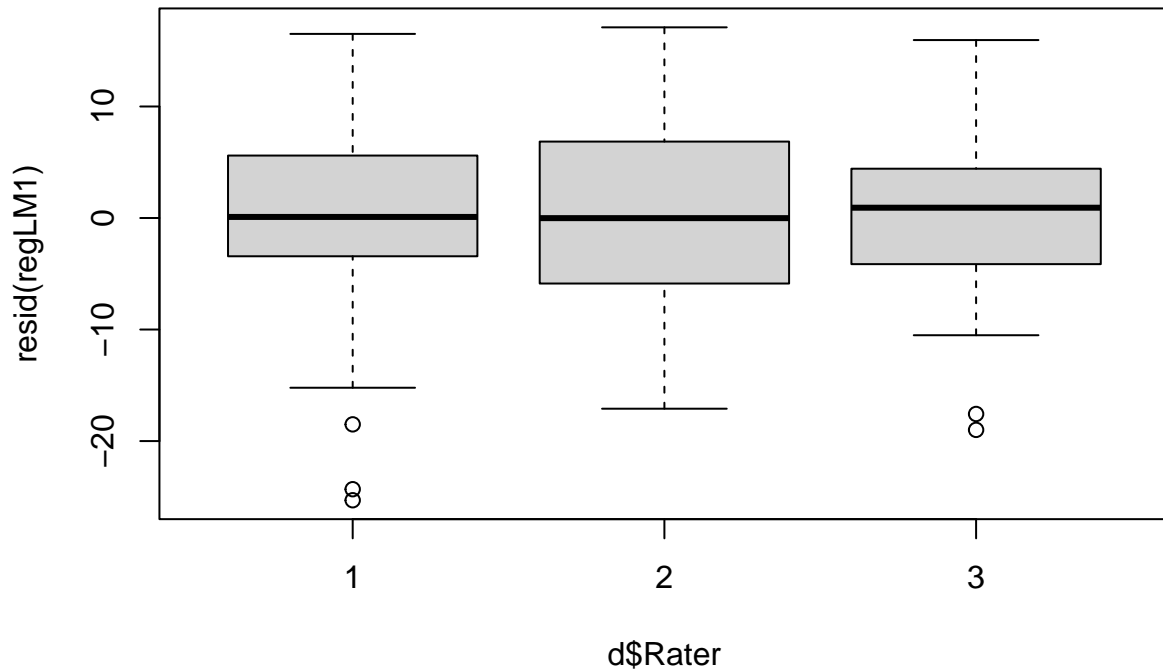
```
# residuals uncorrelated with IVs?  
boxplot(resid(regLM1) ~ d$`Language model`)
```



```
boxplot(resid(regLM1) ~ d$`Is this a prompt chaining case?`)
```



```
boxplot(resid(regLM1) ~ d$Rater)
```



3.2.3 Interaction model (language model x prompt chaining)

```
jtools::summ(regLM1INT <- lmer(SSETtotal ~ `Language model` * `Is this a prompt chaining case?` + Rater
```

```
## MODEL INFO:
## Observations: 180
## Dependent Variable: SSETtotal
## Type: Mixed effects linear regression
##
## MODEL FIT:
## AIC = 1419.85, BIC = 1451.78
## Pseudo-R2 (fixed effects) = 0.56
## Pseudo-R2 (total) = 0.78
##
## FIXED EFFECTS:
```

	Est.	2.5%	97.5%	t val.	d.f.	p
(Intercept)	49.29	43.74	54.84	17.40	96.20	0.00
Language model'Bing Prec.	-9.73	-17.21	-2.24	-2.55	86.17	0.01
Language model'GPT-4	12.85	5.28	20.42	3.33	83.58	0.00
Is this a prompt chaining case?'Yes	-14.23	-23.50	-4.95	-3.01	83.58	0.00
Rater.cat2	16.05	12.16	19.95	8.05	108.63	0.00

```

## Rater.cat3                11.28      7.38   15.18      5.65   108.23   0.00
## Language model'Bing        5.21     -7.76   18.19      0.79    92.89   0.43
## Prec.: 'Is this a prompt
## chaining case?'Yes
## Language model'GPT-4: 'Is  28.07     14.95   41.19      4.19    83.77   0.00
## this a prompt chaining
## case?'Yes
## -----
##
## p values calculated using Kenward-Roger standard errors and d.f.
##
## RANDOM EFFECTS:
## -----
##      Group      Parameter      Std. Dev.
## -----
## Case Code...1 (Intercept)      9.92
## Residual                  10.07
## -----
##
## Grouping variables:
## -----
##      Group      # groups      ICC
## -----
## Case Code...1      90      0.49
## -----

```

reference category: Zero Shot Cases generated by Bard and were rated by Rater 1

Equation:

$$SSET_{total} = 49.29 - 9.73 * Bing + 12.85 * chatGPT - 14.23 * promptchainingYes + 5.21 * Bing * promptchainingYes + 28.07 * chatGPT * promptchainingYes + 16.05 * rater2 + 11.28 * rater3$$

Bard: Zero-shot scenarios scored an average of 49.29 points, as rated by Rater #1. Bard prompt chaining: score decreased by 14.23 points

$$(\beta = -14.23, 95\%CI[-23.50, -4.95], p < 0.001)$$

Bing Precise: Zero-shot scenarios scored 9.73 points lower than Bard's zero-shot scenarios

$$(\beta = -9.73, 95\%CI[-17.21, -2.24], p = 0.01)$$

Bing Precise prompt chaining: score further decreased, but the interaction term was not statistically significant

$$(\beta = 5.21, 95\%CI[-7.76, 18.19], p = 0.43)$$

ChatGPT-4: zero-shot scenarios scored 12.85 points higher than Bard's zero-shot scenarios

$$(\beta = 12.85, 95\%CI[5.28, 20.42], p < 0.001)$$

ChatGPT-4 prompt chaining: score increased significantly by an additional 28.07 points

$$(\beta = 28.07, 95\%CI[14.95, 41.19], p < 0.001)$$

Predicted SSET value for Bing prompt chaining:

```
49.29 - 9.73*1 -14.23*1+5.21*1*1
```

```
## [1] 30.54
```

```
d.regLM1INT <- model.frame(regLM1INT)
names(d.regLM1INT)
```

```
## [1] "SSETtotal" "Language model"
## [3] "Is this a prompt chaining case?" "Rater.cat"
## [5] "Case Code...1"
```

```
d.regLM1INT$SSETtotal <- NULL
d.regLM1INT$`Case Code...1` <- NULL
dim(d.regLM1INT)
```

```
## [1] 180 3
```

```
d.unique <- unique(d.regLM1INT)
dim(d.unique)
```

```
## [1] 18 3
```

```
# d.unique$avgSSET <- predict(regLM1INT, newdata=d.unique)
```

```
d.unique$avgSSET <- NA
```

```
# install.packages("fastDummies")
library(fastDummies)
```

```
## Thank you for using fastDummies!
```

```
## To acknowledge our work, please cite the package:
```

```
## Kaplan, J. & Schlegel, B. (2023). fastDummies: Fast Creation of Dummy (Binary) Columns and Rows from
```

```
d.unique <- dummy_cols(d.unique)
```

```
d.unique$avgSSET <- 49.29 - 9.73*d.unique$`Language model_Bing Prec.` + 12.85*d.unique$`Language model_
```

```
d.unique.rater1 <- d.unique %>% filter(Rater.cat_1==1)
```

```
library(tidyr)
```

```
##
```

```
## Attaching package: 'tidyr'
```

```
## The following objects are masked from 'package:Matrix':
```

```
##
```

```
## expand, pack, unpack
```

```
summaryResult <- d.unique %>% select(`Language model`, `Is this a prompt chaining case?`, Rater.cat, avgSSET)
summaryResult <- summaryResult %>% dplyr::rename(rater1SSET="1", rater2SSET="2", rater3SSET="3")
summaryResult$avgSSET <- (summaryResult$rater1SSET + summaryResult$rater2SSET + summaryResult$rater3SSET)/3
# names(summaryResult)
summaryResult <- summaryResult %>% select(`Language model`, `Is this a prompt chaining case?`, rater1SSET, rater2SSET, rater3SSET, avgSSET)
summaryResult
```

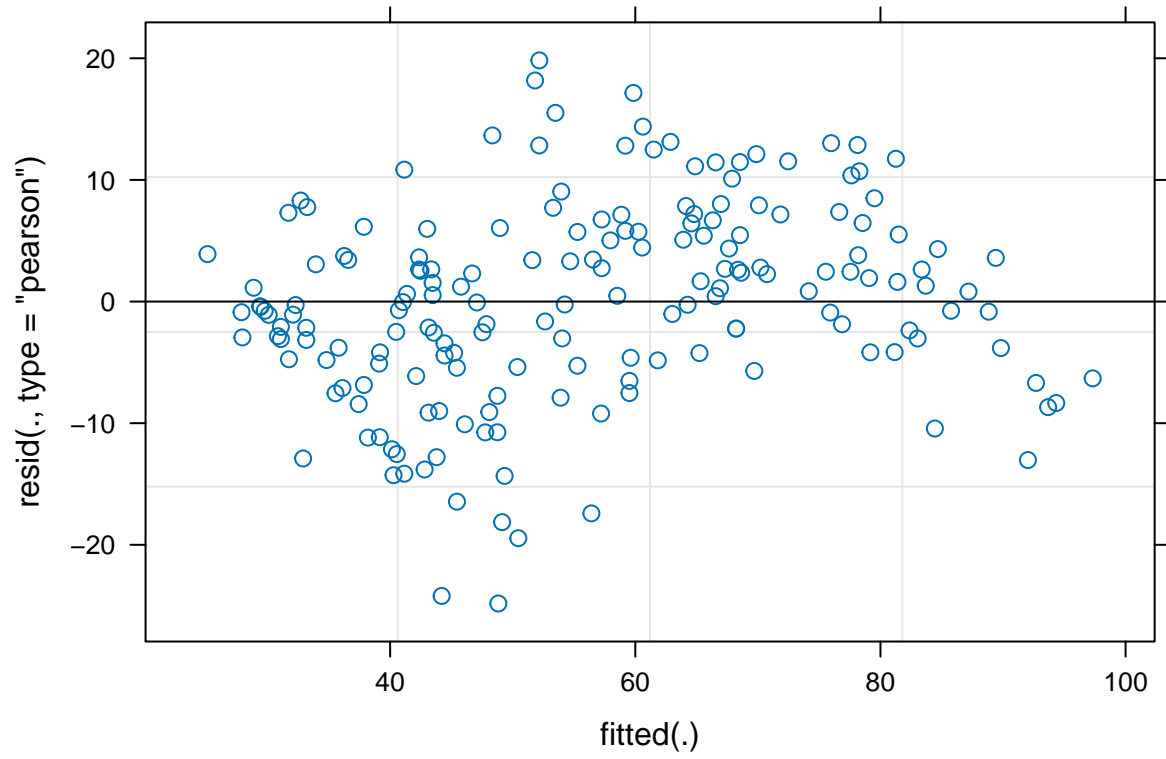
```
## # A tibble: 6 x 6
##   'Language model' Is this a prompt chaining ~1 rater1SSET rater2SSET rater3SSET
##   <fct>           <fct>           <dbl>         <dbl>         <dbl>
## 1 GPT-4          No             62.1          78.2          73.4
## 2 GPT-4          Yes            76.0          92.0          87.3
## 3 Bard           No             49.3          65.3          60.6
## 4 Bard           Yes            35.1          51.1          46.3
## 5 Bing Prec.     No             39.6          55.6          50.8
## 6 Bing Prec.     Yes            30.5          46.6          41.8
## # i abbreviated name: 1: 'Is this a prompt chaining case?'
## # i 1 more variable: avgSSET <dbl>
```

#Diagnostic Tests

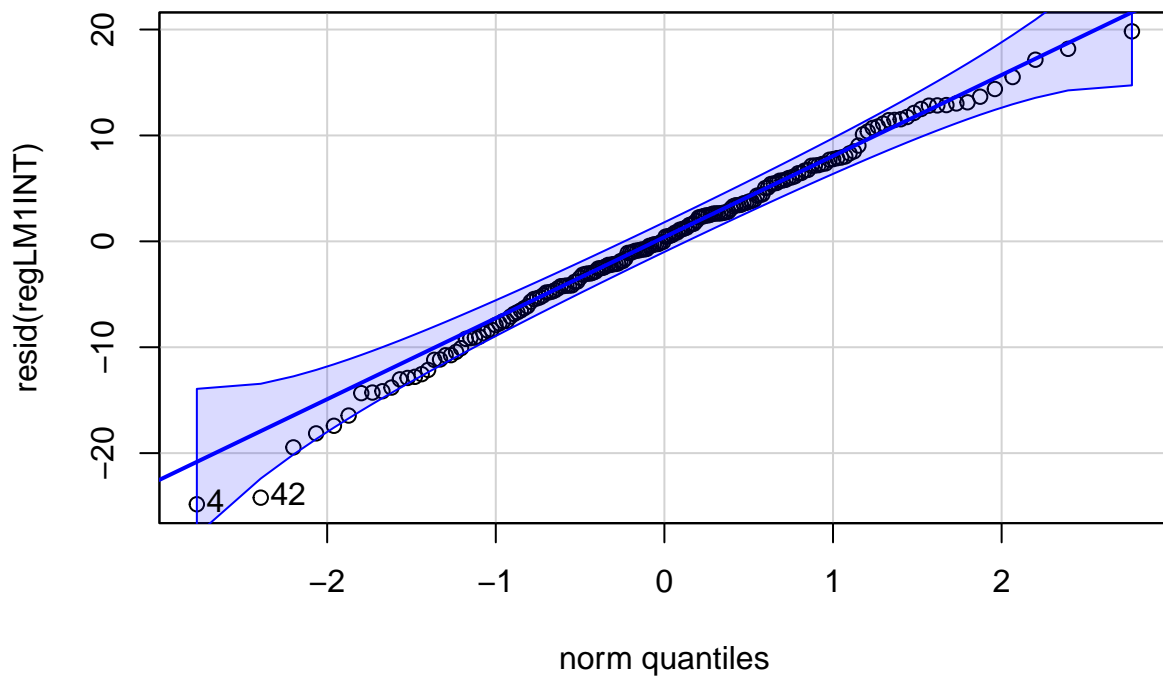
```
library(car)
vif(regLM1INT)
```

```
##                                     GVIF Df GVIF^(1/(2*Df))
## 'Language model'                   2.181897 2      1.215370
## 'Is this a prompt chaining case?'  3.047455 1      1.745696
## Rater.cat                          1.008501 2      1.002118
## 'Language model': 'Is this a prompt chaining case?' 5.182314 2      1.508797
```

```
plot(regLM1INT)
```



```
car::qqPlot(resid(regLM1INT))
```

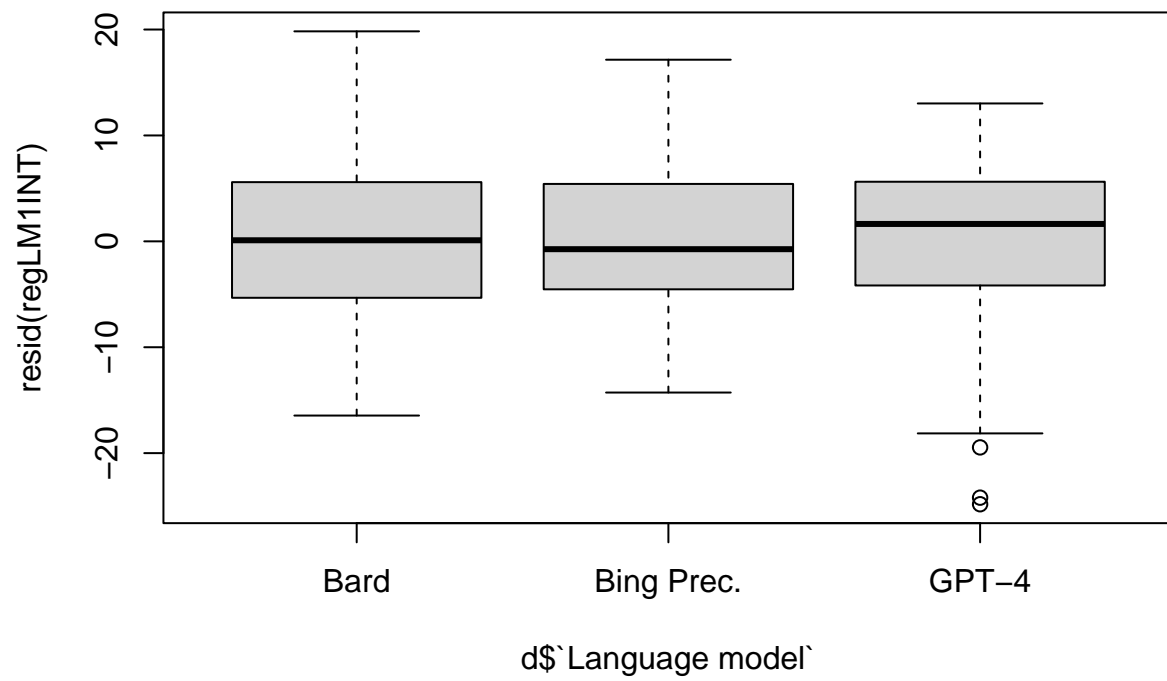



```
## [1] 4 42
```

```
shapiro.test(resid(regLM1INT))
```

```
##
## Shapiro-Wilk normality test
##
## data: resid(regLM1INT)
## W = 0.99266, p-value = 0.5002
```

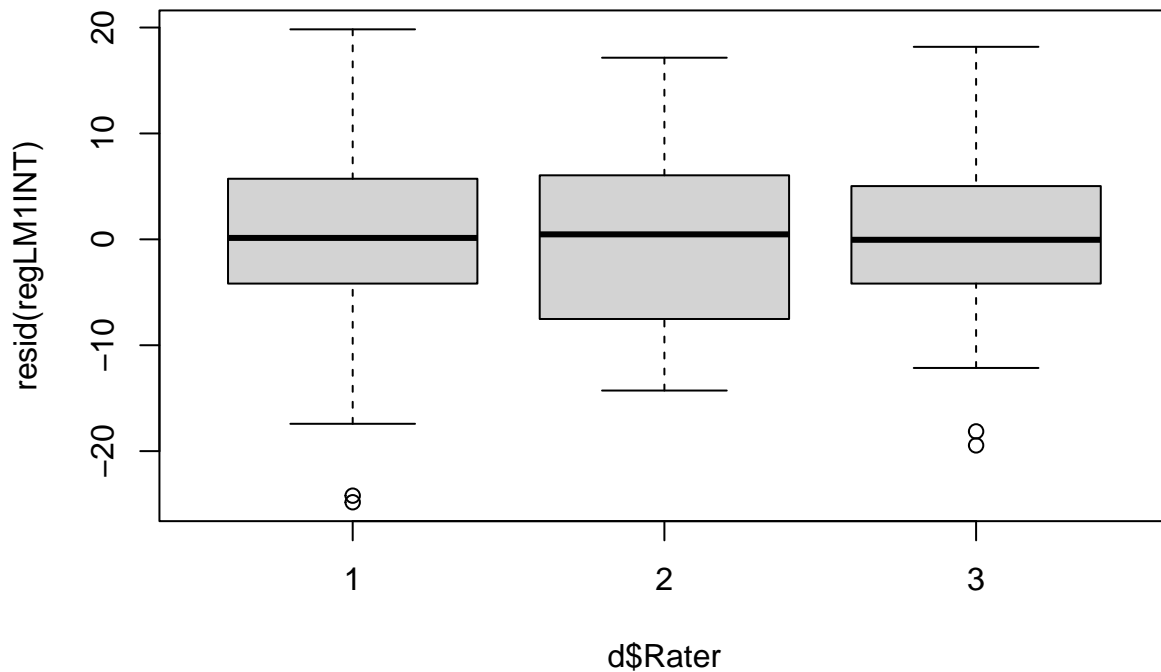
```
# residuals uncorrelated with IVs?
boxplot(resid(regLM1INT) ~ d$`Language model`)
```



```
boxplot(resid(regLM1INT) ~ d$`Is this a prompt chaining case?`)
```



```
boxplot(resid(regLM1INT) ~ d$Rater)
```



```
library(lme4)
library(ggplot2)
library(dplyr)

# Assuming your data is stored in a dataframe called 'd'

# Fit the mixed-effects model
model <- lmer(SSETtotal ~ `Language model` * `Is this a prompt chaining case?` + (1 | Rater), data = d)

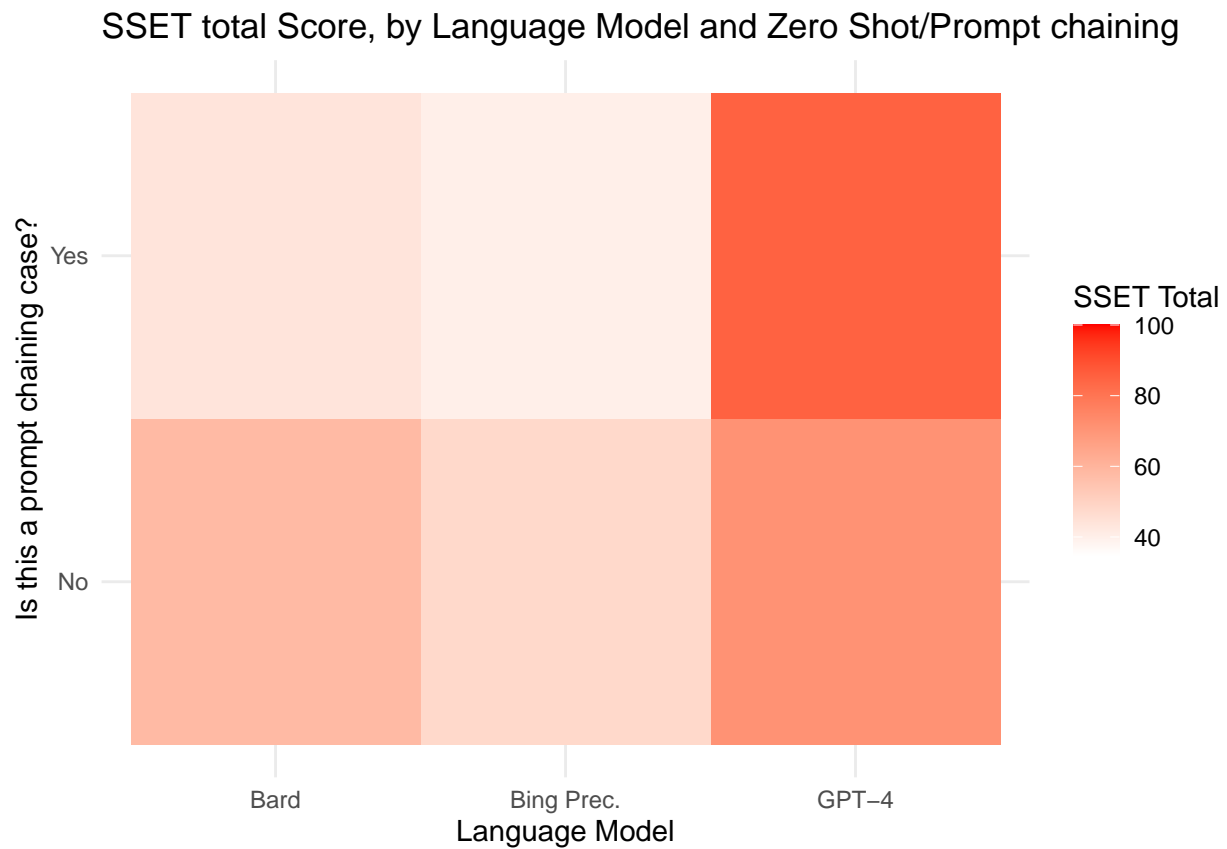
# Generate predicted SSET scores based on the model
d$predicted_sset <- predict(model, newdata = d, re.form = NA)

# Aggregate the predicted SSET scores by Language_model and Prompt Chaining
aggregated_data <- d %>%
  group_by(`Language model`, `Is this a prompt chaining case?`) %>%
  summarise(avg_predicted_sset = mean(predicted_sset))

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

# Create a heatmap
ggplot(aggregated_data, aes(x = `Language model`, y = `Is this a prompt chaining case?`, fill = avg_pre
  geom_tile() +
  scale_fill_gradient(low = "white", high = "red", limit = c(35,100)) +
  labs(title = "SSET total Score, by Language Model and Zero Shot/Prompt chaining",
```

```
x = "Language Model", y = "Is this a prompt chaining case?", fill = "SSET Total") +  
theme_minimal()
```



3.2.4 Interaction model (prompt chaining x rater)

```
jtools::summ(regLM1 <- lmer(SSETtotal ~ `Language model` + as.factor(Rater) * `Is this a prompt chaining
```

```
## MODEL INFO:  
## Observations: 180  
## Dependent Variable: SSETtotal  
## Type: Mixed effects linear regression  
##
```

```
## MODEL FIT:  
## AIC = 1427.58, BIC = 1459.51  
## Pseudo-R2 (fixed effects) = 0.50  
## Pseudo-R2 (total) = 0.81  
##
```

```
## FIXED EFFECTS:
```

	Est.	2.5%	97.5%	t val.	d.f.	p
(Intercept)	44.61	39.07	50.16	15.77	103.33	0.00
Language model'Bing Prec.	-8.16	-15.12	-1.21	-2.30	85.58	0.02

```
## Language model'GPT-4          22.53    15.57    29.49     6.35    85.56    0.00
## as.factor(Rater)2             21.32    16.85    25.79     9.31   101.75    0.00
## as.factor(Rater)3             10.34     5.87    14.81     4.52   101.75    0.00
## Is this a prompt chaining     -0.04    -6.67     6.59    -0.01   133.39    0.99
## case?'Yes
## as.factor(Rater)2:'Is this a  -14.73   -22.58    -6.89    -3.67   102.64    0.00
## prompt chaining case?'Yes
## as.factor(Rater)3:'Is this a    1.17    -6.59     8.93     0.30   100.48    0.77
## prompt chaining case?'Yes
## -----
##
## p values calculated using Kenward-Roger standard errors and d.f.
##
## RANDOM EFFECTS:
## -----
##      Group      Parameter      Std. Dev.
## -----
## Case Code...1 (Intercept)    12.06
## Residual                  9.29
## -----
##
## Grouping variables:
## -----
##      Group      # groups      ICC
## -----
## Case Code...1      90      0.63
## -----
```

- Rater 2 rated zero-shot cases 21.30 points higher than Rater 1, but Rater 2 rated prompt chaining cases 14.69 points *lower* than Rater 1.

3.2.5 Interaction model (language model x rater)

Interaction model to see if raters vary in how they rate cases from each language model, OLS with lm function:

```
summary(regLM1 <- lm(SSETtotal ~ `Language model` * as.factor(Rater) + `Is this a prompt chaining case?`

##
## Call:
## lm(formula = SSETtotal ~ 'Language model' * as.factor(Rater) +
##     'Is this a prompt chaining case?', data = d)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -49.169  -6.726  -0.308   10.459   32.026
##
## Coefficients:
##                                Estimate Std. Error t value
## (Intercept)                   44.902     2.873   15.629
## 'Language model'Bing Prec.      -8.384     3.901   -2.149
## 'Language model'GPT-4          24.267     3.901    6.221
## as.factor(Rater)2              13.200     4.777    2.763
```

```
## as.factor(Rater)3                16.067      4.777      3.363
## 'Is this a prompt chaining case?'Yes      -2.507      2.414     -1.038
## 'Language model'Bing Prec.:as.factor(Rater)2    7.168      6.929      1.034
## 'Language model'GPT-4:as.factor(Rater)2        0.781      6.625      0.118
## 'Language model'Bing Prec.:as.factor(Rater)3   -4.612      6.628     -0.696
## 'Language model'GPT-4:as.factor(Rater)3       -9.350      6.931     -1.349
##                                     Pr(>|t|)
## (Intercept)                               < 2e-16 ***
## 'Language model'Bing Prec.                 0.033055 *
## 'Language model'GPT-4                     3.71e-09 ***
## as.factor(Rater)2                         0.006355 **
## as.factor(Rater)3                         0.000952 ***
## 'Is this a prompt chaining case?'Yes        0.300526
## 'Language model'Bing Prec.:as.factor(Rater)2 0.302404
## 'Language model'GPT-4:as.factor(Rater)2      0.906298
## 'Language model'Bing Prec.:as.factor(Rater)3 0.487455
## 'Language model'GPT-4:as.factor(Rater)3      0.179170
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.11 on 170 degrees of freedom
## Multiple R-squared:  0.5044, Adjusted R-squared:  0.4781
## F-statistic: 19.22 on 9 and 170 DF,  p-value: < 2.2e-16
```

Repeat with LMER, using lmer function:

```
jtools::summ(regLM1 <- lmer(SSETtotal ~ `Language model` * as.factor(Rater) + `Is this a prompt chaining
```

```
## MODEL INFO:
## Observations: 180
## Dependent Variable: SSETtotal
## Type: Mixed effects linear regression
##
## MODEL FIT:
## AIC = 1429.01, BIC = 1467.33
## Pseudo-R2 (fixed effects) = 0.49
## Pseudo-R2 (total) = 0.77
##
## FIXED EFFECTS:
```

	Est.	2.5%	97.5%	t val.	d.f.	p
(Intercept)	45.23	39.49	50.97	15.44	126.04	0.00
Language model'Bing Prec.	-8.42	-16.07	-0.77	-2.16	130.50	0.03
Language model'GPT-4	24.27	16.62	31.91	6.22	130.53	0.00
as.factor(Rater)2	14.71	7.88	21.54	4.20	102.30	0.00
as.factor(Rater)3	14.56	7.73	21.39	4.16	102.30	0.00
Is this a prompt chaining case?'Yes	-3.50	-9.28	2.28	-1.18	92.92	0.24
Language model'Bing Prec.:as.factor(Rater)2	6.21	-3.77	16.19	1.22	104.00	0.23
Language model'GPT-4:as.factor(Rater)2	-0.45	-9.86	8.96	-0.09	100.86	0.93

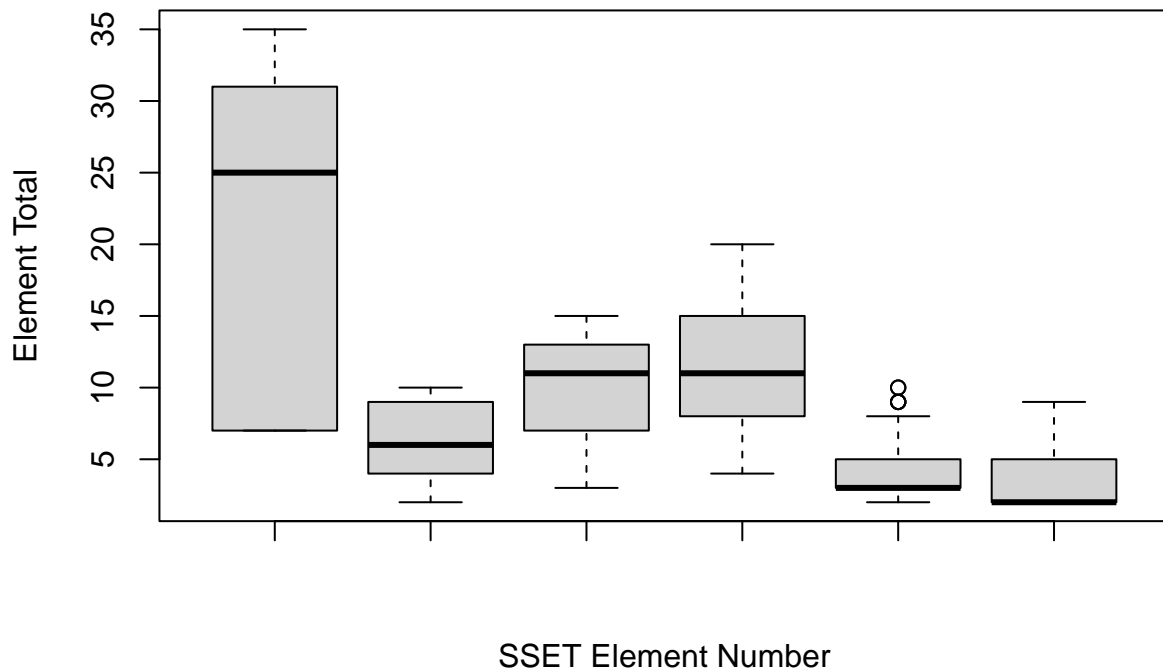
```
## Language model'Bing                -3.47   -12.88    5.94   -0.72   100.80    0.47
## Prec.:as.factor(Rater)3
## Language                -8.20   -18.18    1.78   -1.60   103.81    0.11
## model'GPT-4:as.factor(Rater)3
## -----
##
## p values calculated using Kenward-Roger standard errors and d.f.
##
## RANDOM EFFECTS:
## -----
##      Group      Parameter      Std. Dev.
## -----
## Case Code...1 (Intercept)    11.21
## Residual              10.13
## -----
##
## Grouping variables:
## -----
##      Group      # groups      ICC
## -----
## Case Code...1      90      0.55
## -----
```

4 Search for outliers which consistently scored poorly

Research question: Are there specific cases which consistently scored low on specific SSET items (such as a score of 1 or 2)?

```
boxplot(d$`E1 Total` , d$`E 2 Total` , d$`E3 Total` , d$`E4 Total` , d$`E5 Total` , d$`E6 total` , main =
        xlab="SSET Element Number",
        ylab="Element Total")
```


Distribution of SSET Element Totals



5 Compare the quality of individual SSET elements across the LLM's

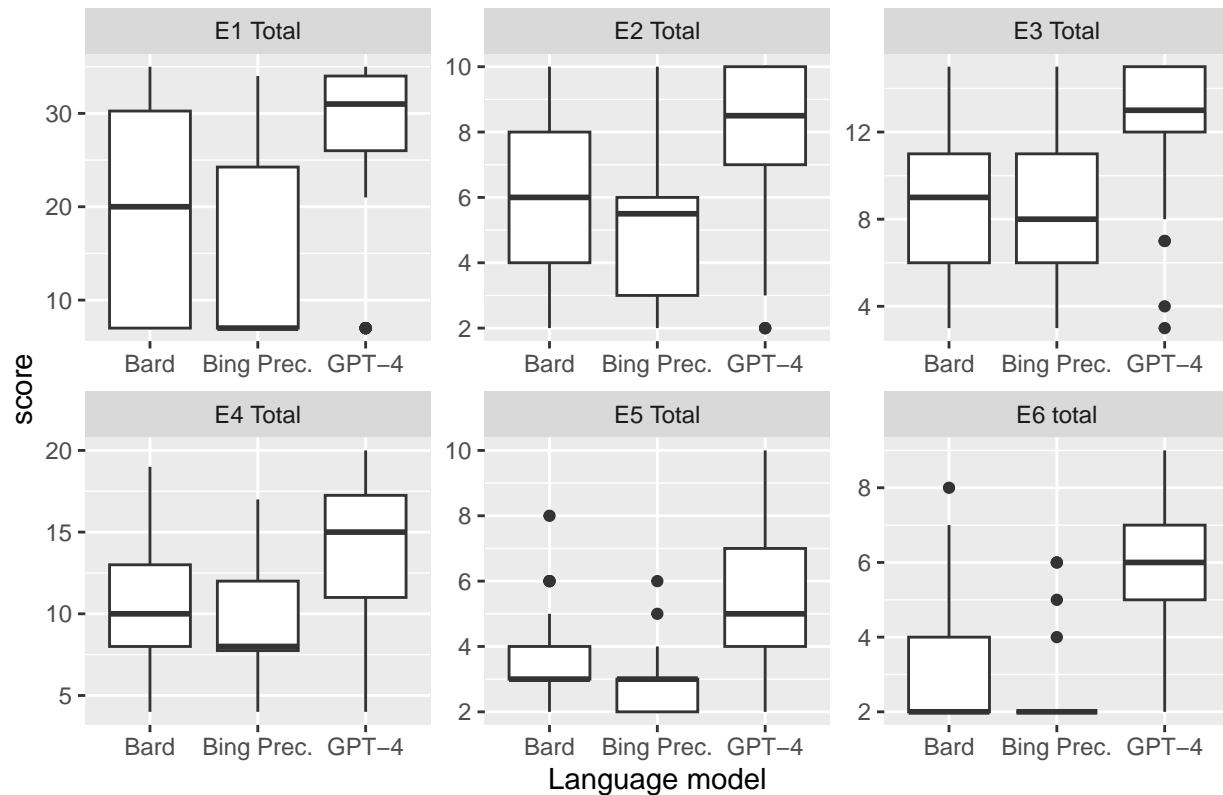
Research question: Do certain LLMs produce higher quality outputs for specific elements of a case?

```
dbox <- d %>% rename(`E2 Total`=`E 2 Total`) %>% select(`E1 Total` , `E2 Total` , `E3 Total` , `E4 Total` , `E5 Total` , `E6 Total`)
dbox <- dbox %>% pivot_longer(cols=c(`E1 Total` , `E2 Total` , `E3 Total` , `E4 Total` , `E5 Total` , `E6 Total`), names_to="element", values_to="score")
names(dbox)
```

```
## [1] "Language model" "element" "score"
```

```
library(ggplot2)
ggplot(dbox, aes(x=`Language model`, y = score))+geom_boxplot() +
  facet_wrap(~dbox$element, scales="free")+
  ggtitle("Disbritubion of Individual SSET Element Scores by Language Model (Two Cases Per Chart)")
```

Disbtribution of Individual SSET Element Scores by Language Model (Two C



```
dboxmean <- d %>%
  group_by(`Case Code...1`) %>%
  summarise(
    `Element 1 Mean` = mean(`E1 Total`),
    `Element 2 Mean` = mean(`E 2 Total`),
    `Element 3 Mean` = mean(`E3 Total`),
    `Element 4 Mean` = mean(`E4 Total`),
    `Element 5 Mean` = mean(`E5 Total`),
    `Element 6 Mean` = mean(`E6 total`),
    `Language model` = first(`Language model`)
  ) %>%
  ungroup()

dim(dboxmean)
```

```
## [1] 90 8
```

```
dboxmeanlong <- dboxmean %>% pivot_longer(cols=c(`Element 1 Mean` , `Element 2 Mean` , `Element 3 Mean`
```

```
library(ggplot2)
ggplot(dboxmeanlong, aes(x=`Language model`, y = score))+geom_boxplot() +
  facet_wrap(~dboxmeanlong$element, scales="free")+
  ggtitle("Disbtribution of Individual SSET Element Scores by Language model")
```

Disbritubion of Individual SSET Element Scores by Language model

