Solution for the assignment of the seventh class

Kovacs Marton

9/29/2021

Importing data

```
processed <- read_tsv("data/boldog_processed.tsv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   .default = col_double(),
##   neme = col_character(),
##   isk = col_character()
## )
## i Use `spec()` for the full column specifications.
```

Data exploration

```
skimr::skim(processed) %>%
  kable()
```

skim_type	skim_variable	n_missing	complete_rate	character.min	character.max	character.empty	character.n_unique	character.whitespace	numeric.mean	numeric.sd	numeric.p0	numeric.p25	numeric.p50	numeric.p75	numeric.p100	numeric.hist
character	neme	0	1.000	2	5	0	2	0	NA	NA	NA	NA	NA	NA	NA	NA
character	isk	0	1.000	7	11	0	4	0	NA	NA	NA	NA	NA	NA	NA	NA
numeric	index	0	1.000	NA	NA	NA	NA	NA	972.34[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	<U+2585><U+2585><U
numeric	eletkora	0	1.000	NA	NA	NA	NA	NA	52.522	[illegible]	[illegible]	45.0000	52.5000	61.0000	[illegible]	<U+2581><U+2585><U
numeric	gyermeke	0	1.000	NA	NA	NA	NA	NA	1.750	[illegible]	[illegible]	1.0000	2.0000	2.00000	9	<U+2585><U+2587><U
numeric	anyagi	0	1.000	NA	NA	NA	NA	NA	3.1000	[illegible]	[illegible]	3.0000	3.0000	3.00000	5	<U+2581><U+2581><U
numeric	pic_elmeny_percent	0	1.000	NA	NA	NA	NA	NA	58.460	[illegible]	[illegible]	40.0000	60.0000	80.00000	100	<U+2582><U+2585><U
numeric	testi_fi	3	0.994	NA	NA	NA	NA	NA	4.1649	[illegible]	[illegible]	4.0000	4.0000	5.00000	6	<U+2581><U+2583><U
numeric	alt_lelki	4	0.992	NA	NA	NA	NA	NA	4.1612	[illegible]	[illegible]	4.0000	4.0000	5.00000	6	<U+2582><U+2583><U
numeric	alt_eg_all	7	0.986	NA	NA	NA	NA	NA	4.1217	[illegible]	[illegible]	3.0000	4.0000	5.00000	6	<U+2582><U+2583><U
numeric	fizero	6	0.988	NA	NA	NA	NA	NA	4.1538	[illegible]	[illegible]	3.0000	4.0000	5.00000	6	<U+2582><U+2585><U
numeric	arcocska	0	1.000	NA	NA	NA	NA	NA	2.5700	[illegible]	[illegible]	2.0000	2.0000	3.00000	7	<U+2587><U+2585><U
numeric	aggodalo	0	1.000	NA	NA	NA	NA	NA	2.7720	[illegible]	[illegible]	2.0000	2.5000	4.00000	6	<U+2587><U+2583><U
numeric	ideges	0	1.000	NA	NA	NA	NA	NA	2.4900	[illegible]	[illegible]	1.0000	2.0000	3.00000	6	<U+2587><U+2582><U
numeric	feszult	0	1.000	NA	NA	NA	NA	NA	2.6100	[illegible]	[illegible]	1.0000	2.0000	3.00000	6	<U+2587><U+2583><U
numeric	nyugtala	0	1.000	NA	NA	NA	NA	NA	2.4060	[illegible]	[illegible]	1.0000	2.0000	3.00000	6	<U+2587><U+2582><U

skim_type	skim_variable	n_missing	complete_rate	character.min	character.max	character.empty	character.n_unique	character.whitespace	numeric.mean	numeric.sd	numeric.p0	numeric.p25	numeric.p50	numeric.p75	numeric.p100	numeric.hist
numeric	diener1	0	1.000	NA	NA	NA	NA	NA	5.5140013286590			5.00006.0000		6.000000	7	<U+2581><U+2581><U
numeric	diener2	0	1.000	NA	NA	NA	NA	NA	5.2280013219346			4.00006.0000		6.000000	7	<U+2581><U+2581><U
numeric	diener3	0	1.000	NA	NA	NA	NA	NA	5.5740012310788			5.00006.0000		6.000000	7	<U+2581><U+2581><U
numeric	diener4	0	1.000	NA	NA	NA	NA	NA	5.4180012611332			5.00006.0000		6.000000	7	<U+2581><U+2581><U
numeric	diener5	0	1.000	NA	NA	NA	NA	NA	5.5760010727374			5.00006.0000		7.000000	7	<U+2581><U+2581><U
numeric	diener6	0	1.000	NA	NA	NA	NA	NA	5.7680011245902			5.00006.0000		7.000000	7	<U+2581><U+2581><U
numeric	diener7	0	1.000	NA	NA	NA	NA	NA	5.6200013738176			5.00006.0000		7.000000	7	<U+2581><U+2581><U
numeric	diener8	0	1.000	NA	NA	NA	NA	NA	5.5320012148777			5.00006.0000		6.000000	7	<U+2581><U+2581><U
numeric	jollet	0	1.000	NA	NA	NA	NA	NA	4.4886166108404			4.00004.6667		5.333300	6	<U+2581><U+2582><U
numeric	savor	0	1.000	NA	NA	NA	NA	NA	4.5146170133860			4.00004.6667		5.000000	6	<U+2581><U+2582><U
numeric	a_vhat	0	1.000	NA	NA	NA	NA	NA	4.5956009393964			4.00004.8000		5.200000	6	<U+2581><U+2581><U
numeric	oreg	0	1.000	NA	NA	NA	NA	NA	4.2173322217368			3.33334.3333		5.083325	6	<U+2582><U+2583><U
numeric	rezil	0	1.000	NA	NA	NA	NA	NA	4.0446170210453			3.33334.0000		4.666700	6	<U+2581><U+2583><U
numeric	m_flow	0	1.000	NA	NA	NA	NA	NA	4.7739978909798			4.33335.0000		5.333300	6	<U+2581><U+2581><U
numeric	g_jerz	0	1.000	NA	NA	NA	NA	NA	4.1528100313775			3.40004.2000		5.000000	6	<U+2581><U+2583><U
numeric	g_jpszi	0	1.000	NA	NA	NA	NA	NA	4.2820098928403			3.75004.5000		5.000000	6	<U+2581><U+2582><U
numeric	g_jszoc	0	1.000	NA	NA	NA	NA	NA	3.9495101789209			3.00004.0000		5.000000	6	<U+2582><U+2585><U
numeric	g_jspir	0	1.000	NA	NA	NA	NA	NA	4.2455102003167			3.50004.2500		5.250000	6	<U+2582><U+2583><U
numeric	g_jerzpsz	0	1.000	NA	NA	NA	NA	NA	4.2174009720234			3.61874.3375		4.925000	6	<U+2581><U+2582><U
numeric	g_jspszoc	0	1.000	NA	NA	NA	NA	NA	4.0975101382101			3.37504.1250		5.000000	6	<U+2581><U+2583><U
numeric	pk_mm	0	1.000	NA	NA	NA	NA	NA	3.0273346267487			2.66673.1667		3.500000	4	<U+2581><U+2582><U
numeric	pk_av	0	1.000	NA	NA	NA	NA	NA	3.1750006232189			2.75003.2500		3.750000	4	<U+2581><U+2581><U
numeric	pk_onr	0	1.000	NA	NA	NA	NA	NA	3.0640007912331			2.66673.3333		3.666700	4	<U+2581><U+2582><U
numeric	pk_rez	0	1.000	NA	NA	NA	NA	NA	3.1506687424924			2.66673.3333		3.666700	4	<U+2581><U+2582><U
numeric	p_poz_erz	0	1.000	NA	NA	NA	NA	NA	21.7425027510389			20.000023.0000		26.000030	0	<U+2581><U+2581><U
numeric	p_elmely	0	1.000	NA	NA	NA	NA	NA	22.3604055858401			20.000023.0000		26.000030	0	<U+2581><U+2582><U
numeric	p_poz_kapc	0	1.000	NA	NA	NA	NA	NA	22.4425056710017			19.000024.0000		26.000030	0	<U+2581><U+2581><U
numeric	p_ert_cel	0	1.000	NA	NA	NA	NA	NA	23.3744067909301			21.000024.0000		27.000030	0	<U+2581><U+2581><U
numeric	p_telj	0	1.000	NA	NA	NA	NA	NA	22.7164034200099			21.000024.0000		26.000030	0	<U+2581><U+2581><U
numeric	p_boldog	0	1.000	NA	NA	NA	NA	NA	7.4040201217406			7.00008.0000		9.000010		<U+2581><U+2581><U
numeric	p_egeszs	0	1.000	NA	NA	NA	NA	NA	22.2605022480403			19.000023.0000		26.000030	0	<U+2581><U+2581><U
numeric	p_neg_erz	0	1.000	NA	NA	NA	NA	NA	9.7660507089779			5.00009.0000		13.000030	0	<U+2587><U+2587><U
numeric	p_magany	0	1.000	NA	NA	NA	NA	NA	3.1620305655407			0.00002.0000		5.000010		<U+2587><U+2582><U
numeric	perma	0	1.000	NA	NA	NA	NA	NA	169.3731026721131			155.00017050		191.000225	00	<U+2581><U+2581><U

1. Look for a variable thats variance is explained by the 8 Diener items by at least 65%.

I will first calculate the total score for the Diener flourishing scale.

```
diener_data <-
  processed %>%
  mutate(diener_sum = diener1 + diener2 + diener3 + diener4 + diener5 + diener6 + diener7 + diener8)
```

For this task I am investigating only interval variables. To calculate the Rsquared I calculate the Pearson correlation coefficient and square it.

```
interval_vars <-
  processed %>%
```

```r
  select(26:50) %>%
  names()

rsq <- function (data, x, y) cor(data[[x]], data[[y]]) ^ 2

diener_res <-
  tibble::tibble(
    variable = interval_vars,
    rsquared = map_dbl(variable,
                  ~ rsq(diener_data, .x, "diener_sum"))
    ) %>%
  mutate(
    rsquared = round(rsquared, 2),
    flag = case_when(rsquared >= .65 ~ TRUE,
                     rsquared < .65 ~ FALSE)
  ) %>%
  arrange(desc(rsquared))
```

The total score on the Diener 8 item scale only explains more than 65% of the variance of the *g_jerzpsz* scale.

2. Lets test whether adding aggodalom and ideges variables as main effects will increase the R2.

To do this we have to create a linear regression model.

```r
m <- lm(g_jerzpsz ~ diener_sum + aggodalo + ideges, data = diener_data)

summary(m)
```

```
##
## Call:
## lm(formula = g_jerzpsz ~ diener_sum + aggodalo + ideges, data = diener_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.31675 -0.35127  0.02655  0.40331  2.07871
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.457202   0.157206   2.908   0.0038 **
## diener_sum   0.090846   0.003008  30.201   <2e-16 ***
## aggodalo    -0.044526   0.028036  -1.588   0.1129
## ideges      -0.060587   0.029656  -2.043   0.0416 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5348 on 496 degrees of freedom
## Multiple R-squared:  0.6991, Adjusted R-squared:  0.6973
## F-statistic: 384.1 on 3 and 496 DF,  p-value: < 2.2e-16
```

Adding those two variables indeed increased the Rsquared to 0.6991.

3. Transforming the kor variable

```
kor_data <-
  processed %>%
  dplyr::select(eletkora, index) %>%
  # bind_cols(kor_data, poly(kor_data$eletkora), degree = 3)
  mutate(korz = scale(eletkora),
         korz2 = korz^2,
         korz3 = korz^3,
         korz4 = korz^4)
```

Now, lets create a correlation matrix including these variables.

```
Hmisc::rcorr(as.matrix(kor_data), type = "pearson")
```

```
##          eletkora index  korz korz2 korz3 korz4
## eletkora     1.00  0.00  1.00 -0.09  0.80 -0.14
## index        0.00  1.00  0.00  0.13 -0.04  0.10
## korz         1.00  0.00  1.00 -0.09  0.80 -0.14
## korz2       -0.09  0.13 -0.09  1.00 -0.22  0.88
## korz3        0.80 -0.04  0.80 -0.22  1.00 -0.34
## korz4       -0.14  0.10 -0.14  0.88 -0.34  1.00
##
## n= 500
##
##
## P
##          eletkora index  korz   korz2  korz3  korz4
## eletkora          0.9189 0.0000 0.0501 0.0000 0.0014
## index    0.9189          0.9189 0.0046 0.4237 0.0211
## korz     0.0000   0.9189        0.0501 0.0000 0.0014
## korz2    0.0501   0.0046 0.0501        0.0000 0.0000
## korz3    0.0000   0.4237 0.0000 0.0000        0.0000
## korz4    0.0014   0.0211 0.0014 0.0000 0.0000
```

4. Predicting PERMA by korz with polynomial regression. Which power has the largest effect on the outcome variable? Plot the results.

```
polyreg_data <-
  kor_data %>%
  left_join(., select(processed, perma, index), by = "index")

m <- lm(perma ~ korz + korz2 + korz3 + korz4, data = polyreg_data)

summary(m)
```

```
##
```

```
## Call:
## lm(formula = perma ~ korz + korz2 + korz3 + korz4, data = polyreg_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -152.046  -15.746    6.024   21.075   58.192
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 174.9017     1.9641  89.051  < 2e-16 ***
## korz          6.1223     2.3521   2.603 0.009521 **
## korz2        -8.0878     2.2799  -3.547 0.000426 ***
## korz3        -0.6376     0.7452  -0.856 0.392650
## korz4         0.9206     0.4145   2.221 0.026791 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 30.57 on 495 degrees of freedom
## Multiple R-squared:  0.0516, Adjusted R-squared:  0.04393
## F-statistic: 6.733 on 4 and 495 DF,  p-value: 2.783e-05
```

The second and fourth degree polynomial variables predicted the perma outcome variable significantly.

5. Look for a variable that has a third degree polynomial relationsip with age. Plot the relationship.

To look for the relationship I will look at the correlation between the interval variables and the third degree polynomial of age. I am looking for a significant relationship.

```
third_data <-
  processed %>%
  left_join(., kor_data, by = "index")

third_res <-
  tibble::tibble(
    variable = interval_vars,
    cor_res = map(variable,
                  ~ my_cor(
                    data = third_data,
                    x = .x,
                    y = "korz3",
                    method = "spearman"
                  )),
    cor_r = map_dbl(cor_res, ~ pluck(.x, "estimate", "rho")),
    cor_p = map_dbl(cor_res, ~ pluck(.x, "p.value")),
    flagged = case_when(cor_p <= 0.05 ~ TRUE,
                        TRUE ~ FALSE)
  ) %>%
  arrange(desc(cor_r))
```

```
## Warning in cor.test.default(data[[x]], data[[y]], method = method): Cannot
```

```
## compute exact p-value with ties

## Warning in cor.test.default(data[[x]], data[[y]], method = method): Cannot
## compute exact p-value with ties

## Warning in cor.test.default(data[[x]], data[[y]], method = method): Cannot
## compute exact p-value with ties

## Warning in cor.test.default(data[[x]], data[[y]], method = method): Cannot
## compute exact p-value with ties

## Warning in cor.test.default(data[[x]], data[[y]], method = method): Cannot
## compute exact p-value with ties

## Warning in cor.test.default(data[[x]], data[[y]], method = method): Cannot
## compute exact p-value with ties

## Warning in cor.test.default(data[[x]], data[[y]], method = method): Cannot
## compute exact p-value with ties

## Warning in cor.test.default(data[[x]], data[[y]], method = method): Cannot
## compute exact p-value with ties

## Warning in cor.test.default(data[[x]], data[[y]], method = method): Cannot
## compute exact p-value with ties

## Warning in cor.test.default(data[[x]], data[[y]], method = method): Cannot
## compute exact p-value with ties

## Warning in cor.test.default(data[[x]], data[[y]], method = method): Cannot
## compute exact p-value with ties

## Warning in cor.test.default(data[[x]], data[[y]], method = method): Cannot
## compute exact p-value with ties

## Warning in cor.test.default(data[[x]], data[[y]], method = method): Cannot
## compute exact p-value with ties

## Warning in cor.test.default(data[[x]], data[[y]], method = method): Cannot
## compute exact p-value with ties

## Warning in cor.test.default(data[[x]], data[[y]], method = method): Cannot
## compute exact p-value with ties

## Warning in cor.test.default(data[[x]], data[[y]], method = method): Cannot
## compute exact p-value with ties

## Warning in cor.test.default(data[[x]], data[[y]], method = method): Cannot
## compute exact p-value with ties

## Warning in cor.test.default(data[[x]], data[[y]], method = method): Cannot
## compute exact p-value with ties

## Warning in cor.test.default(data[[x]], data[[y]], method = method): Cannot
```

```
## compute exact p-value with ties

## Warning in cor.test.default(data[[x]], data[[y]], method = method): Cannot
## compute exact p-value with ties

## Warning in cor.test.default(data[[x]], data[[y]], method = method): Cannot
## compute exact p-value with ties

## Warning in cor.test.default(data[[x]], data[[y]], method = method): Cannot
## compute exact p-value with ties

## Warning in cor.test.default(data[[x]], data[[y]], method = method): Cannot
## compute exact p-value with ties

## Warning in cor.test.default(data[[x]], data[[y]], method = method): Cannot
## compute exact p-value with ties

## Warning in cor.test.default(data[[x]], data[[y]], method = method): Cannot
## compute exact p-value with ties
```

The results show that 10 variables had a significant third degree polynomial relationship with the age variable. The following table is in descending order based on the Spearmans' rho.

```
third_res %>%
  select(-cor_res)
```

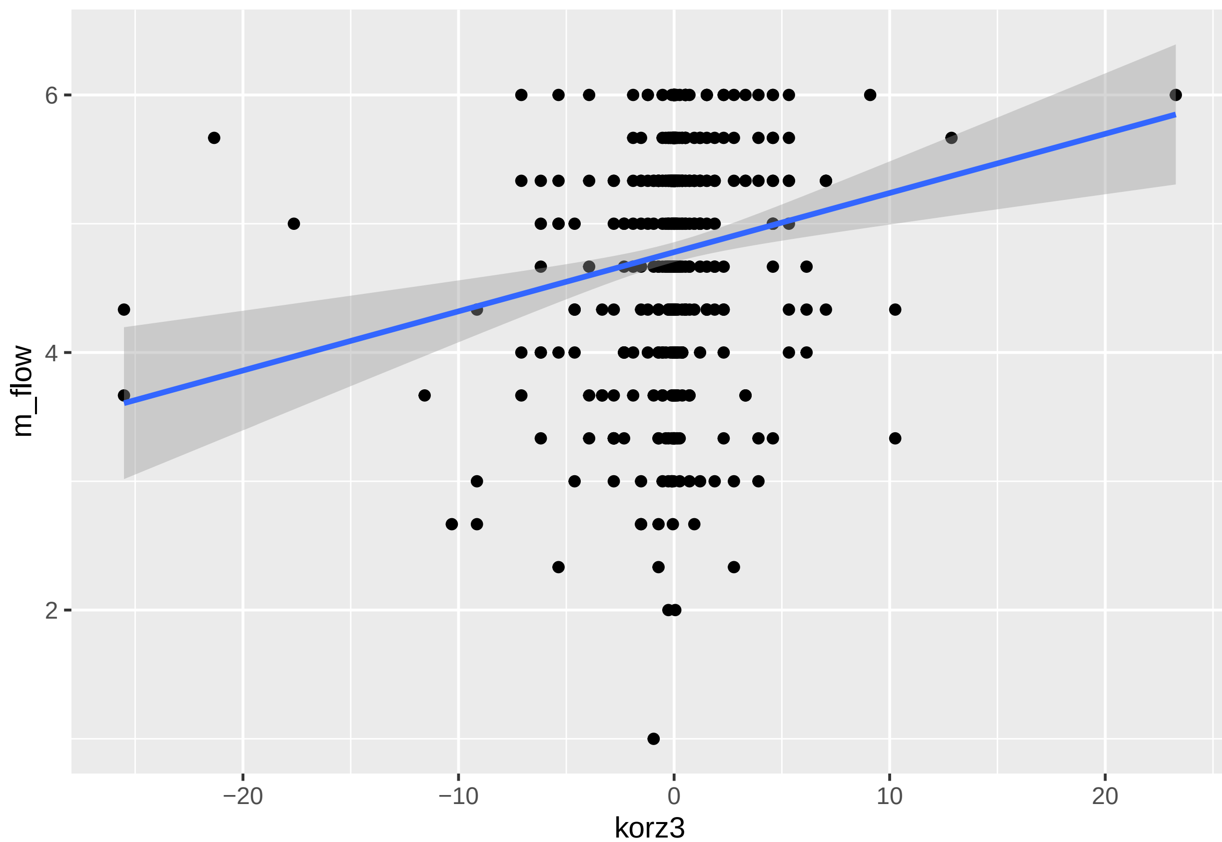
```
## # A tibble: 25 x 4
##    variable  cor_r      cor_p flagged
##    <chr>     <dbl>      <dbl> <lgl>
##  1 m_flow    0.207 0.00000319 TRUE
##  2 onreg     0.188 0.0000243  TRUE
##  3 pik_onr   0.176 0.0000771  TRUE
##  4 p_boldog  0.172 0.000113   TRUE
##  5 p_poz_erz 0.168 0.000158   TRUE
##  6 pik_rez   0.165 0.000216   TRUE
##  7 perma     0.145 0.00115    TRUE
##  8 p_elmely  0.125 0.00505    TRUE
##  9 rezil     0.119 0.00776    TRUE
## 10 g_jerz    0.114 0.0111     TRUE
## # ... with 15 more rows
```

Now, I would like to create scatterplot showing the relationship between these variables and age.

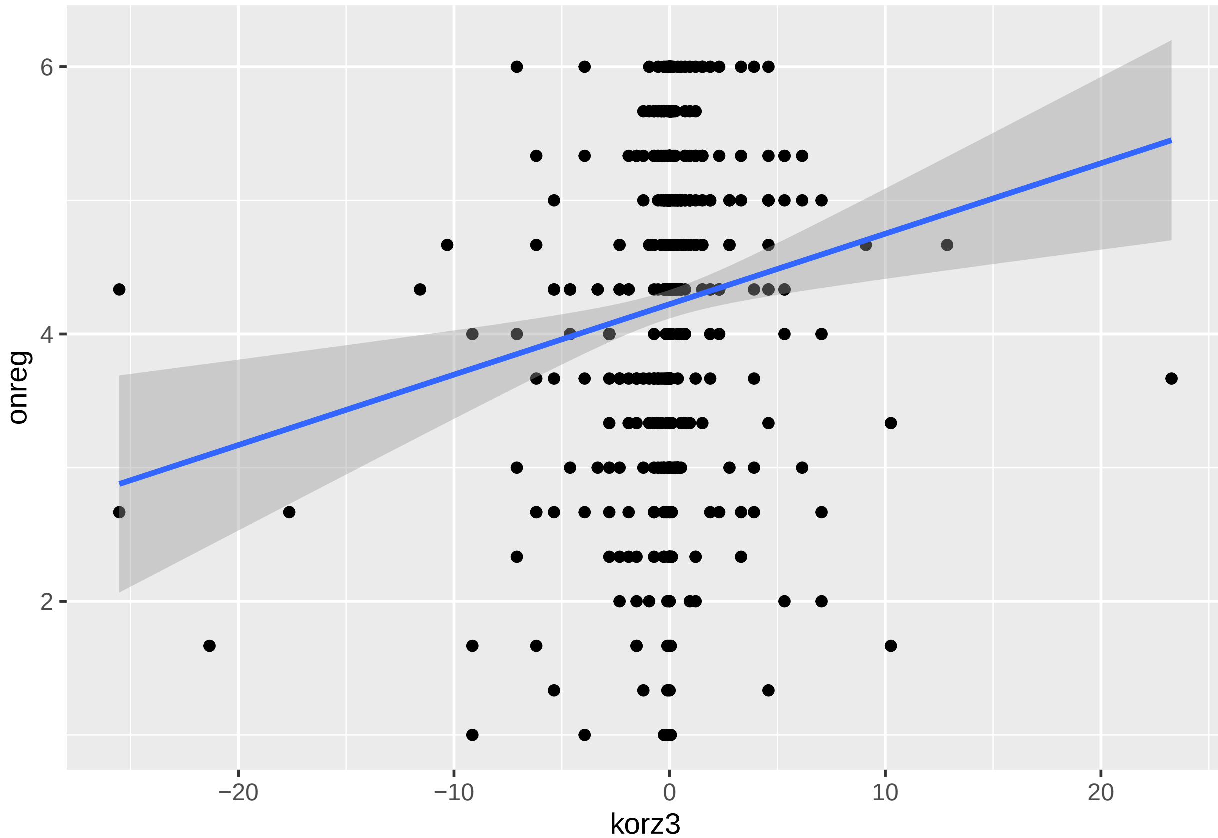
```
third_plot <-
  third_res %>%
  dplyr::filter(flagged) %>%
  mutate(plot = map(variable,
                    ~ poly_plot(
                      data = third_data,
                      x = "korz3",
                      y = .x
                    ))
         )
```

```
third_plot$plot
```

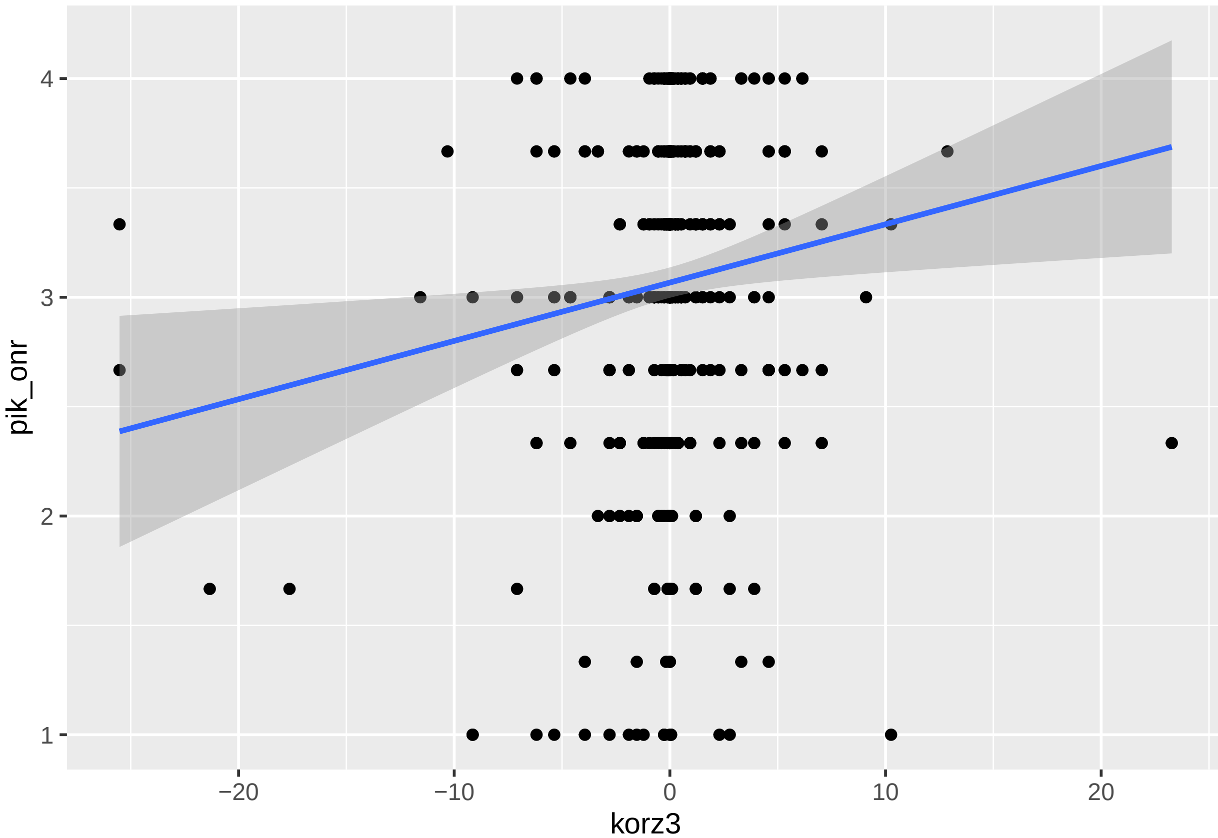
```
## [[1]]
```



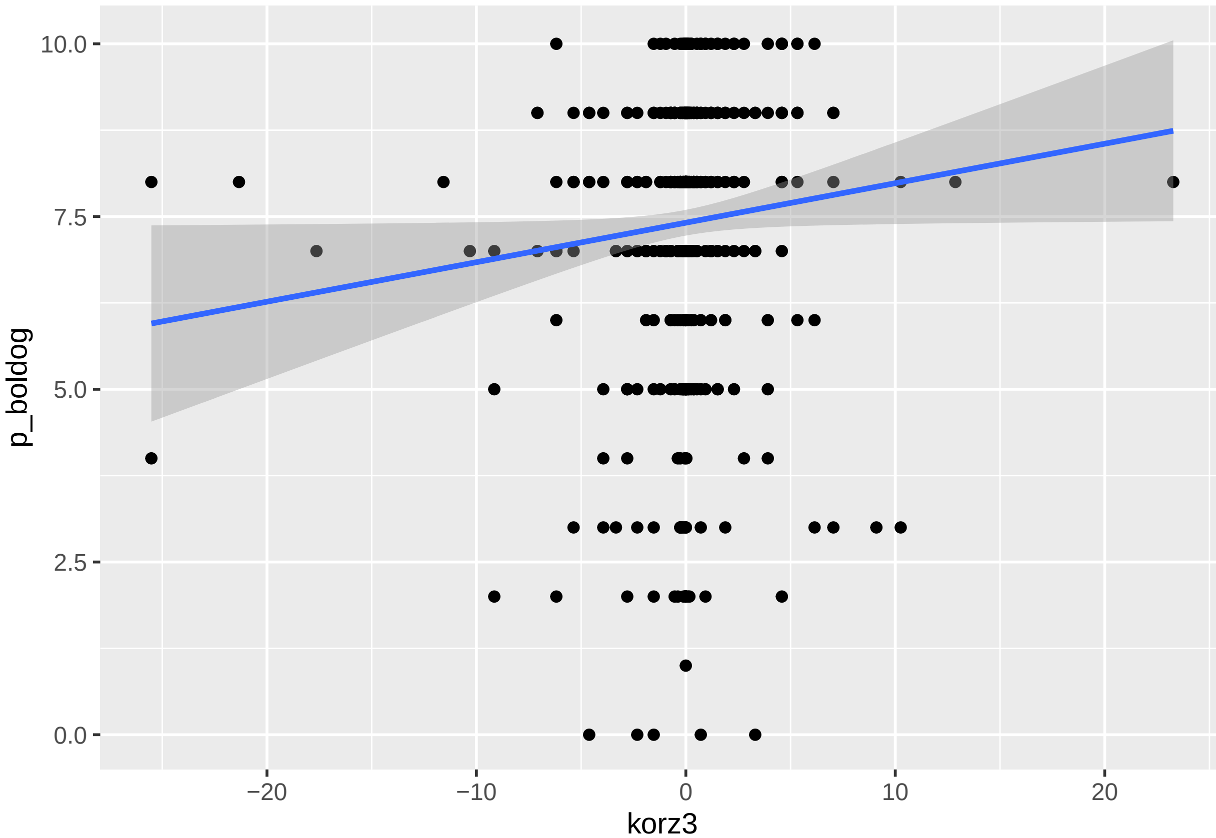

```
##
## [[2]]
```



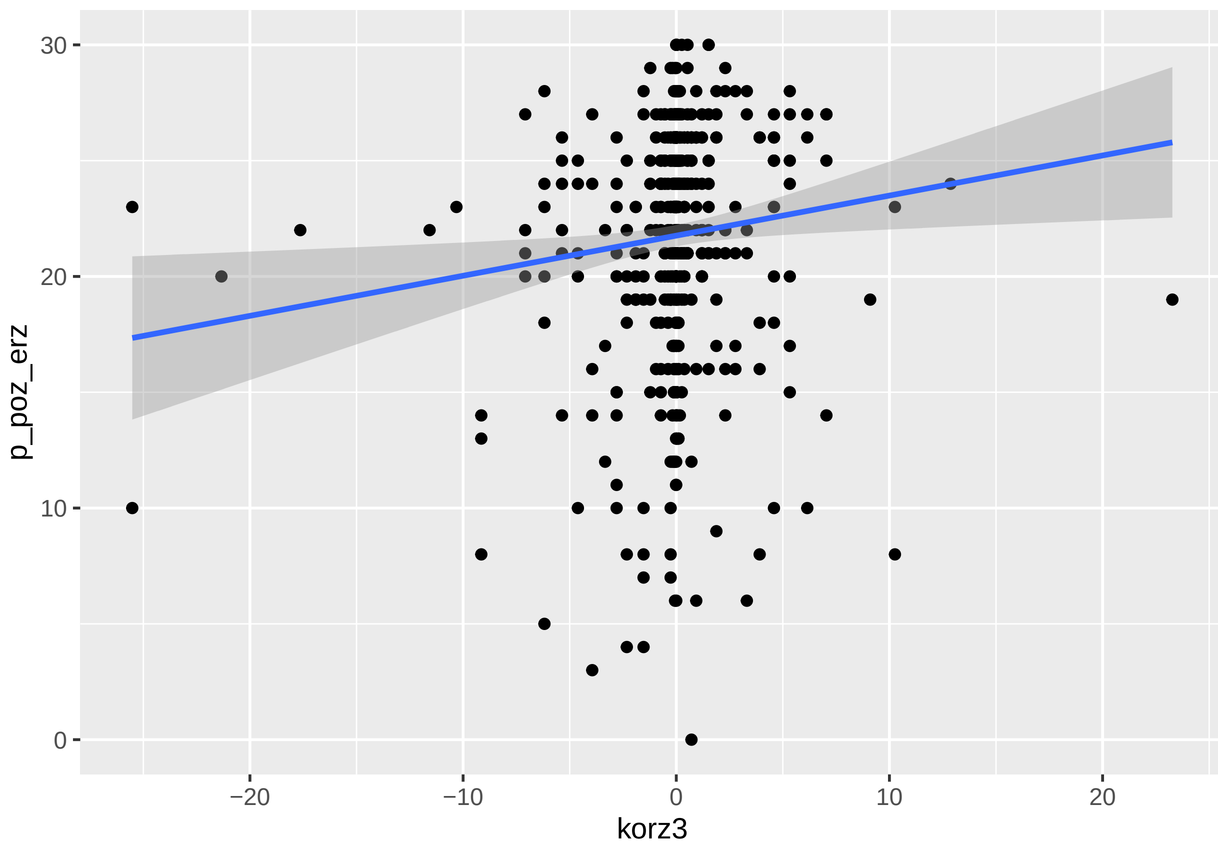

```
##
## [[3]]
```



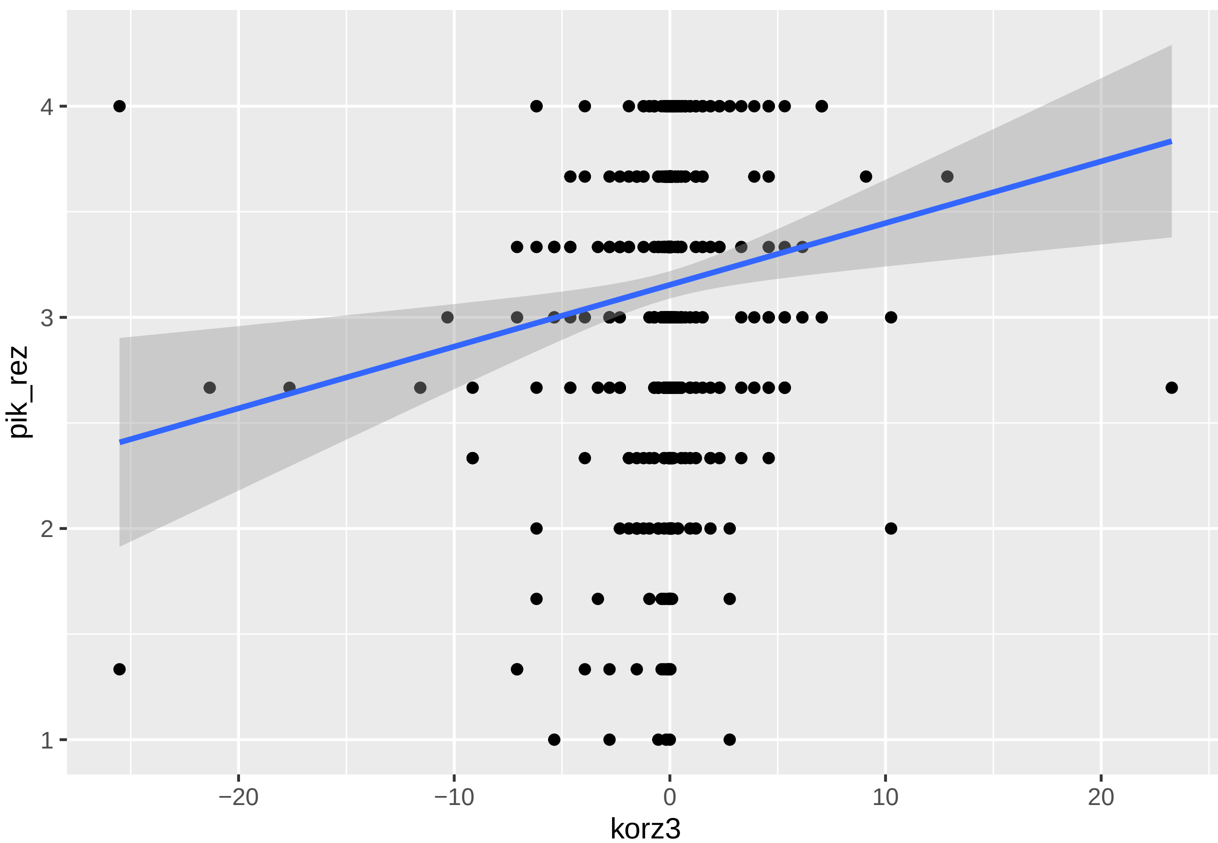

```
##
## [[4]]
```



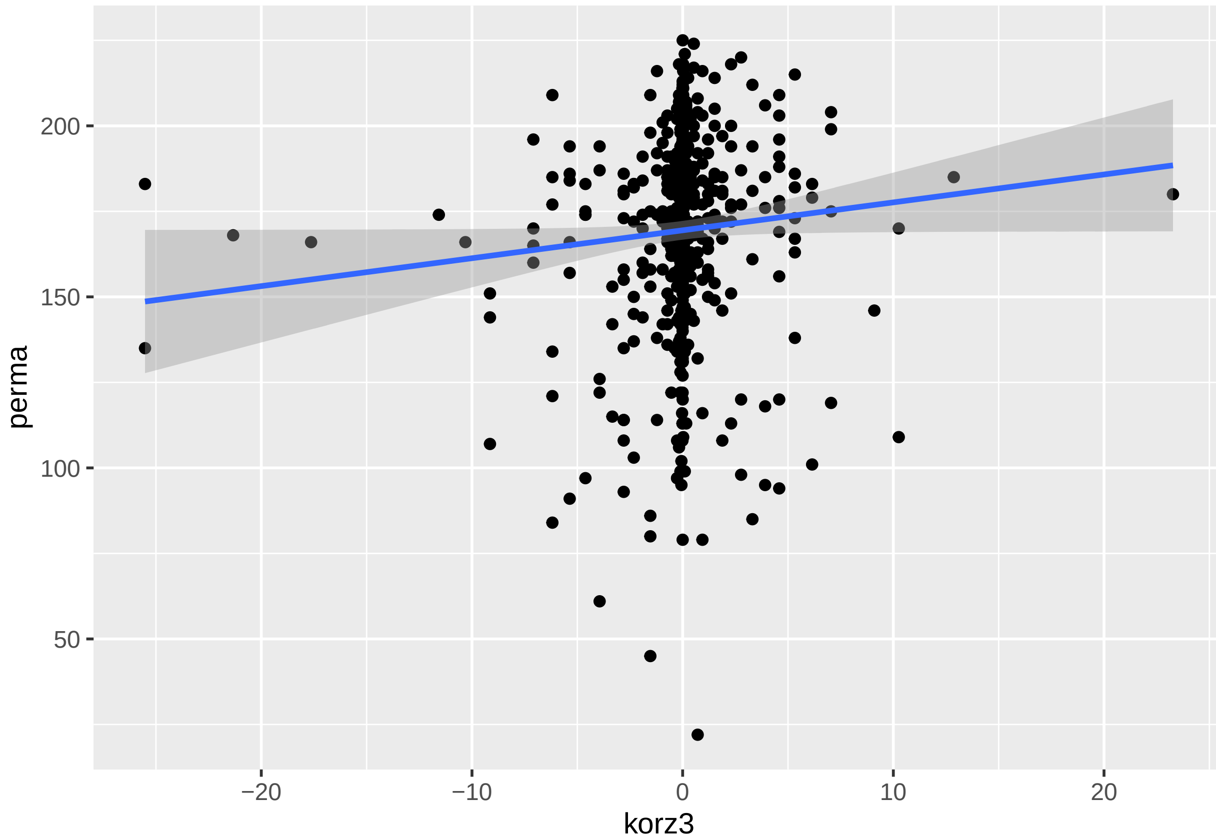

```
##
## [[5]]
```



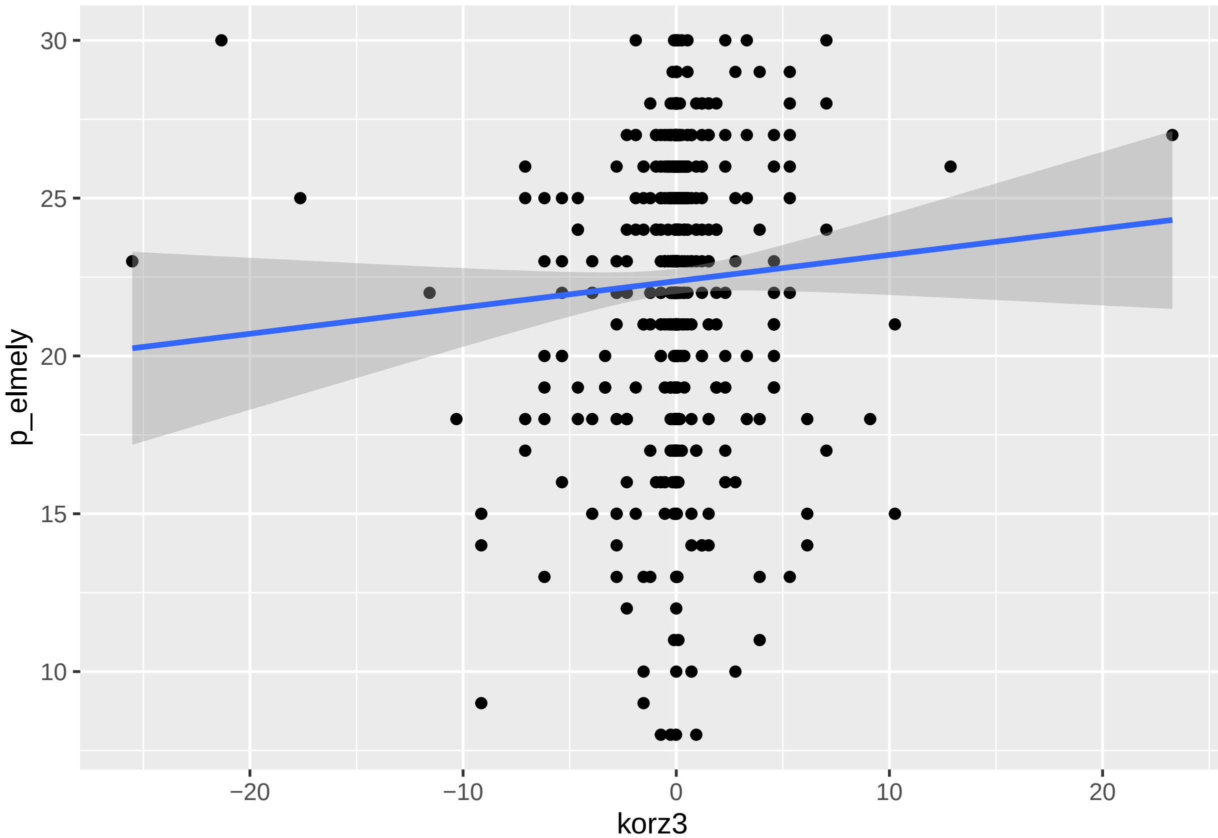

```
##
## [[6]]
```



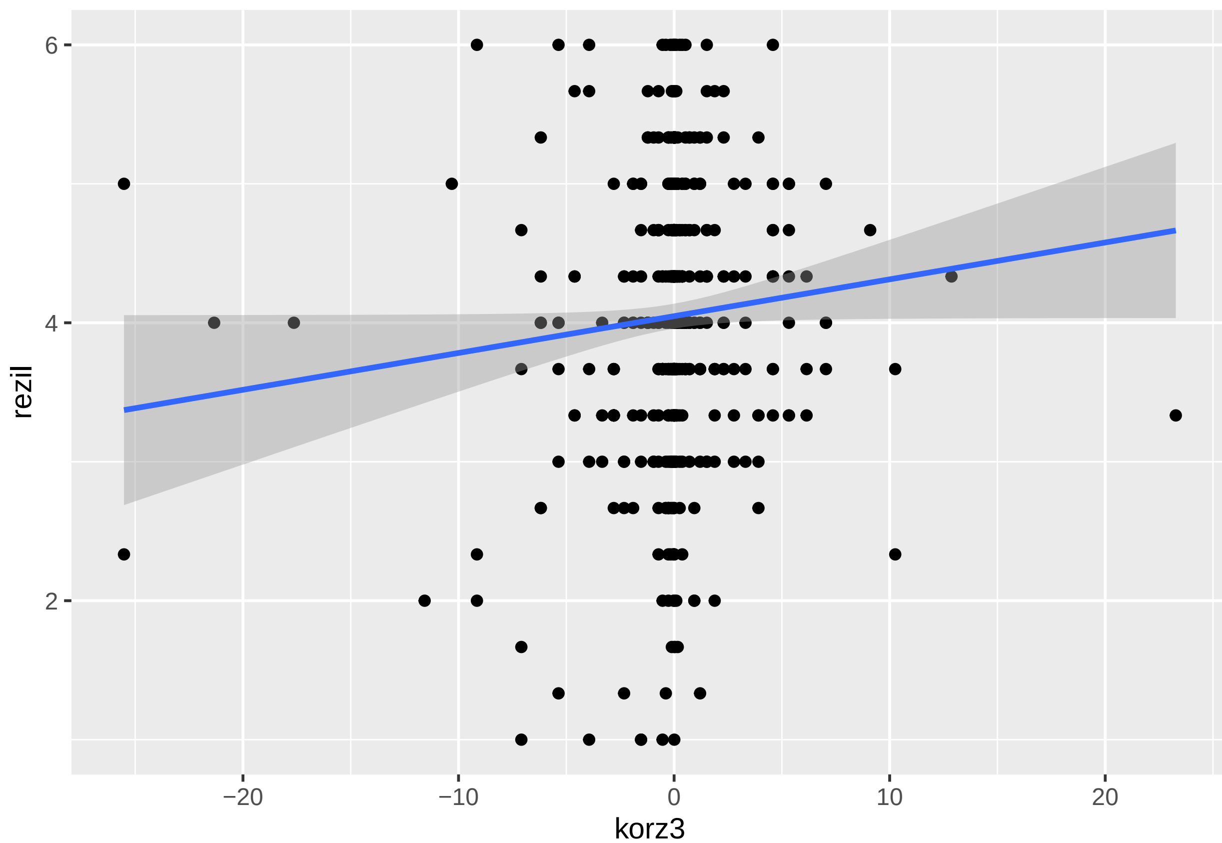

```
##
## [[7]]
```



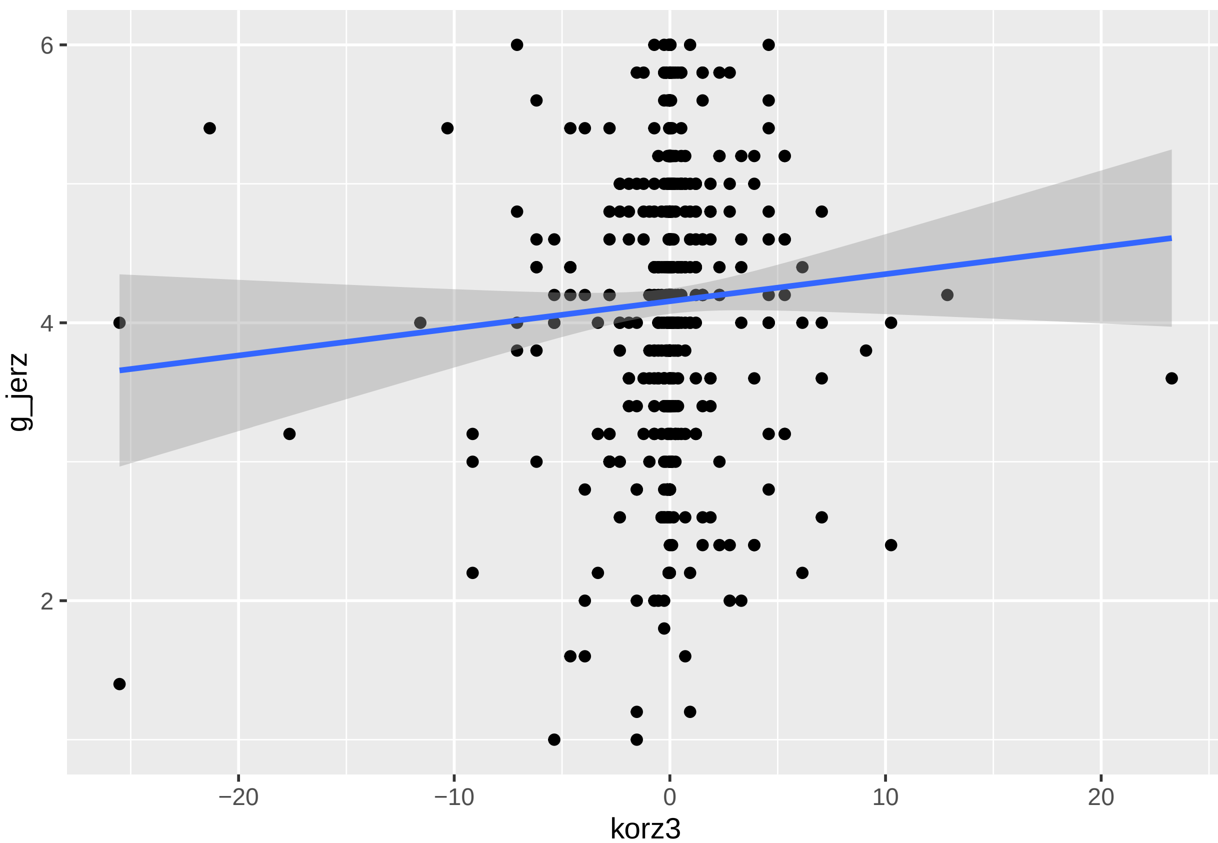

```
##
## [[8]]
```



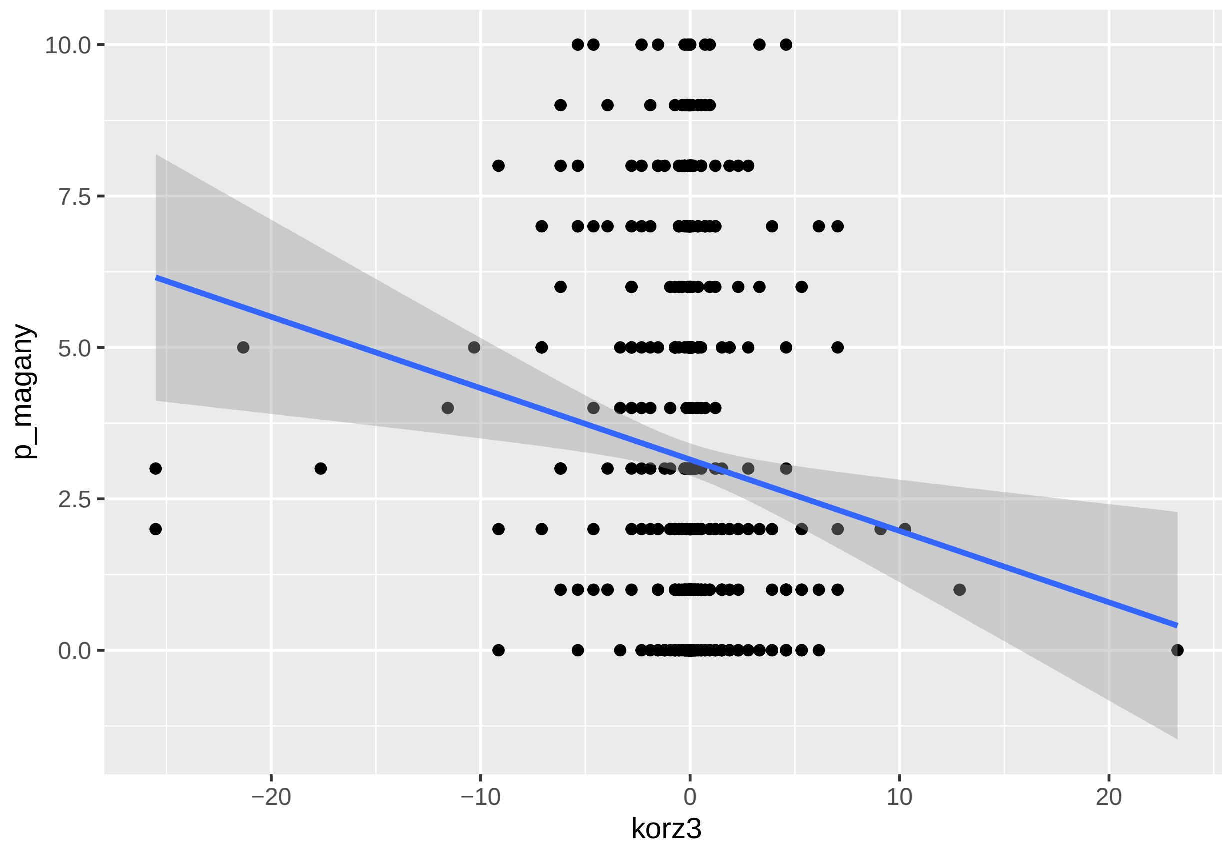

```
##
## [[9]]
```



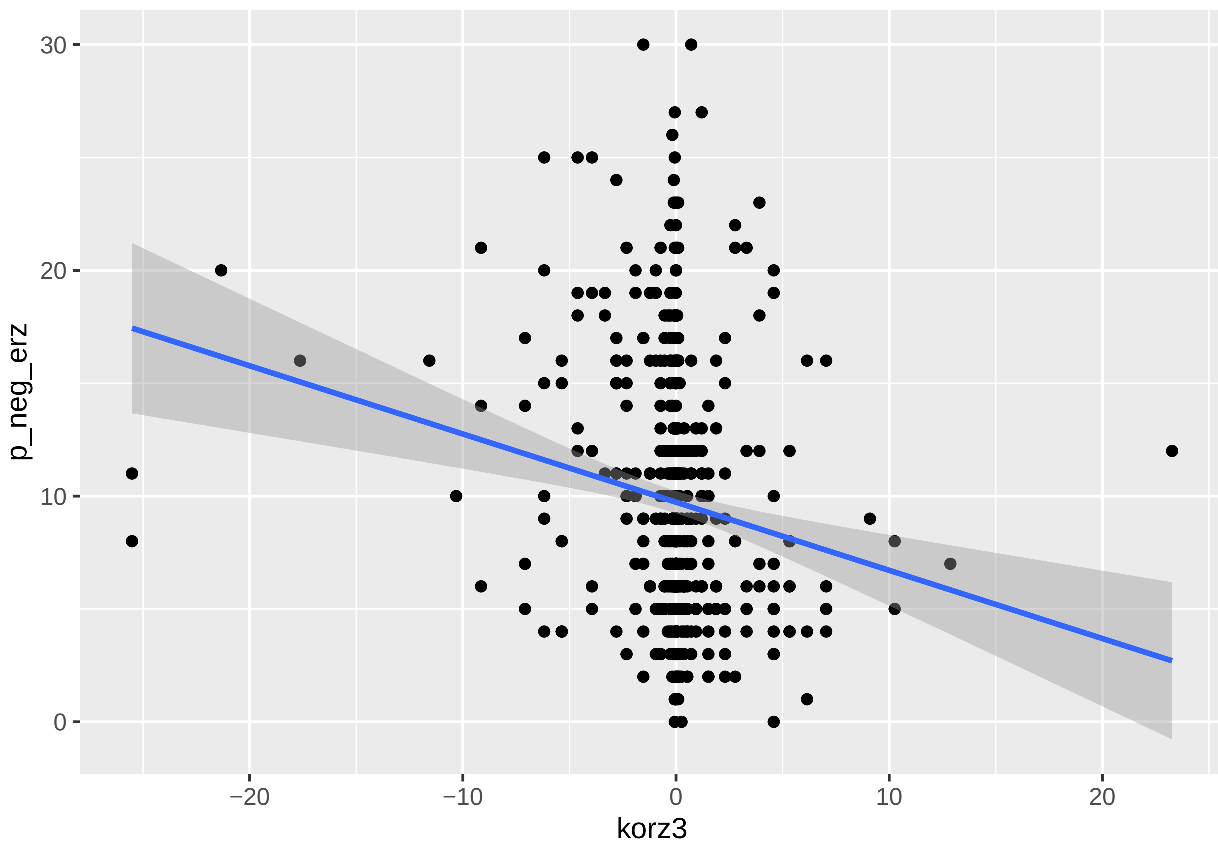

```
##
## [[10]]
```




```
##
## [[11]]
```




```
##
## [[12]]
```



6. Multiple linear regression where pelmeny is the outcome variable and Aggodalom, Ideges, Feszült, Nyugtalan are the predictor variables

```
m <- lm(p_elmeny_percent ~ aggodalo + ideges + feszult + nyugtala, data = processed)

summary(m)
```

```
##
## Call:
## lm(formula = p_elmeny_percent ~ aggodalo + ideges + feszult +
##     nyugtala, data = processed)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -61.397 -12.963   2.806  13.565  46.543
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  79.8519     1.8996  42.036   <2e-16 ***
## aggodalo     -0.3307     1.0691  -0.309   0.7572
## ideges       -3.6558     1.3685  -2.671   0.0078 **
```

```
## feszult      -3.0396     1.4024  -2.167   0.0307 *
## nyugtala     -1.4293     0.7555  -1.892   0.0591 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.52 on 495 degrees of freedom
## Multiple R-squared:  0.2585, Adjusted R-squared:  0.2525
## F-statistic: 43.15 on 4 and 495 DF,  p-value: < 2.2e-16
```

The rsquared suggest that the predictor variables explain 25.85% of the variance in the outcome variable.

For the predictor variables the VIF is the highest for the *feszult* variable.

```
vif <- VIF(m)

vif
```

```
## aggodalo   ideges  feszult nyugtala
## 3.413217 5.149895 5.603635 1.765334
```

And the tolerances are:

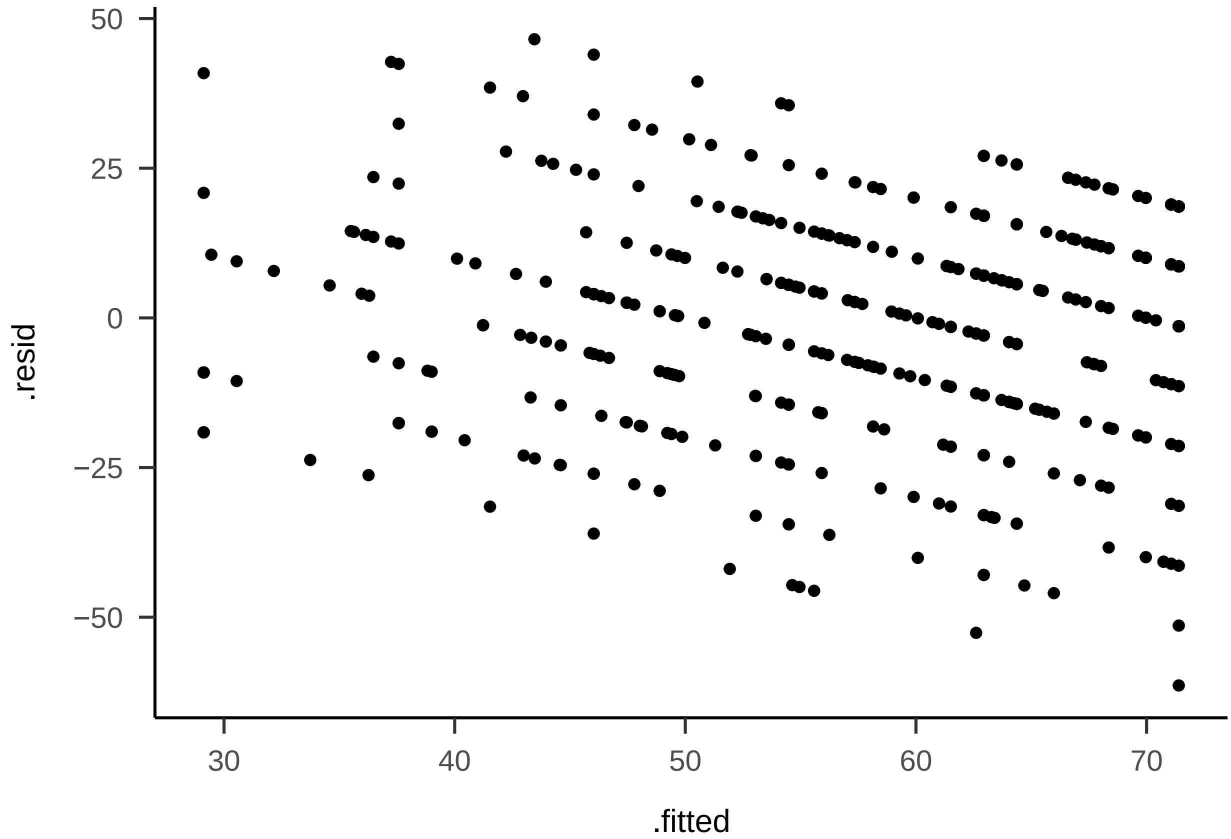
```
1 / vif
```

```
##  aggodalo    ideges   feszult  nyugtala
## 0.2929787 0.1941787 0.1784556 0.5664649
```

Plotting the standardized residuals.

```
to_plot <- broom::augment(m)

ggplot(to_plot,
       aes(x = .fitted, y = .resid)) +
  geom_point() +
  papaja::theme_apa()
```



50
25
0
−25
−50
.resid
30
40
50
60
70
.fitted