Assignment 4 – Solutions: Part 1 (Corruption and Wealth)

Applied Quantitative Methods II, UC3M

1. Setup and data exploration

a) Load the dataset:

```
library(dplyr)
library(broom)
library(ggplot2)
library(modelsummary)
library(marginaleffects)
library(readstata13)

df = read.dta13("https://raw.githubusercontent.com/franvillamil/AQM2/refs/heads/master/datasets/other/corr
```

b) Drop observations with missing values on the key variables:

```
df = df %>% filter(!is.na(ti_cpi) & !is.na(undp_gdp))
nrow(df)
```

```
## [1] 170
```

c) Summary statistics:

```
summary(df$ti_cpi)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.200   2.500   3.300   4.051   4.900   9.700
```

```
sd(df$ti_cpi)
```

```
## [1] 2.105143
```

```
summary(df$undp_gdp)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     520    1974    5280    8950   10862   61190
```

```
sd(df$undp_gdp)
```

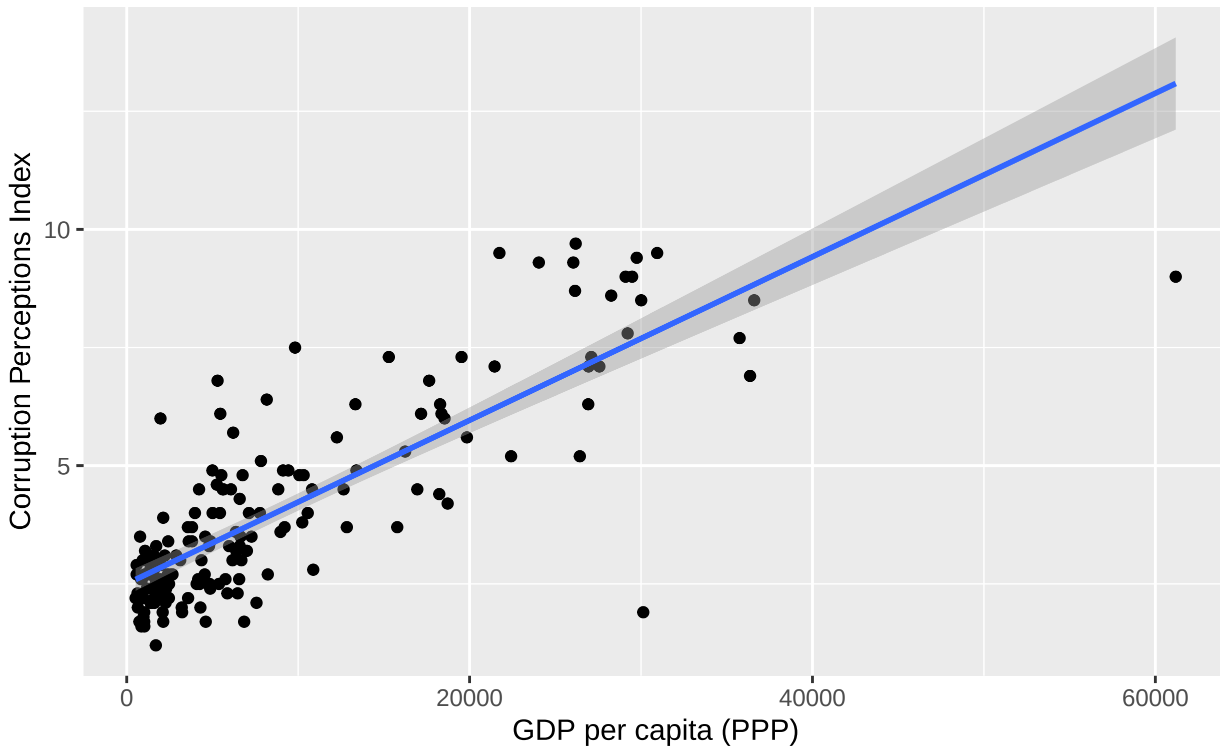
```
## [1] 9986.849
```

The corruption index ranges from its minimum to its maximum on the 0–10 scale. GDP per capita has a large standard deviation relative to its mean and a maximum far above the median, indicating right skewness.

2. Exploratory visualization

a) Scatter plot of corruption vs. GDP per capita (level):

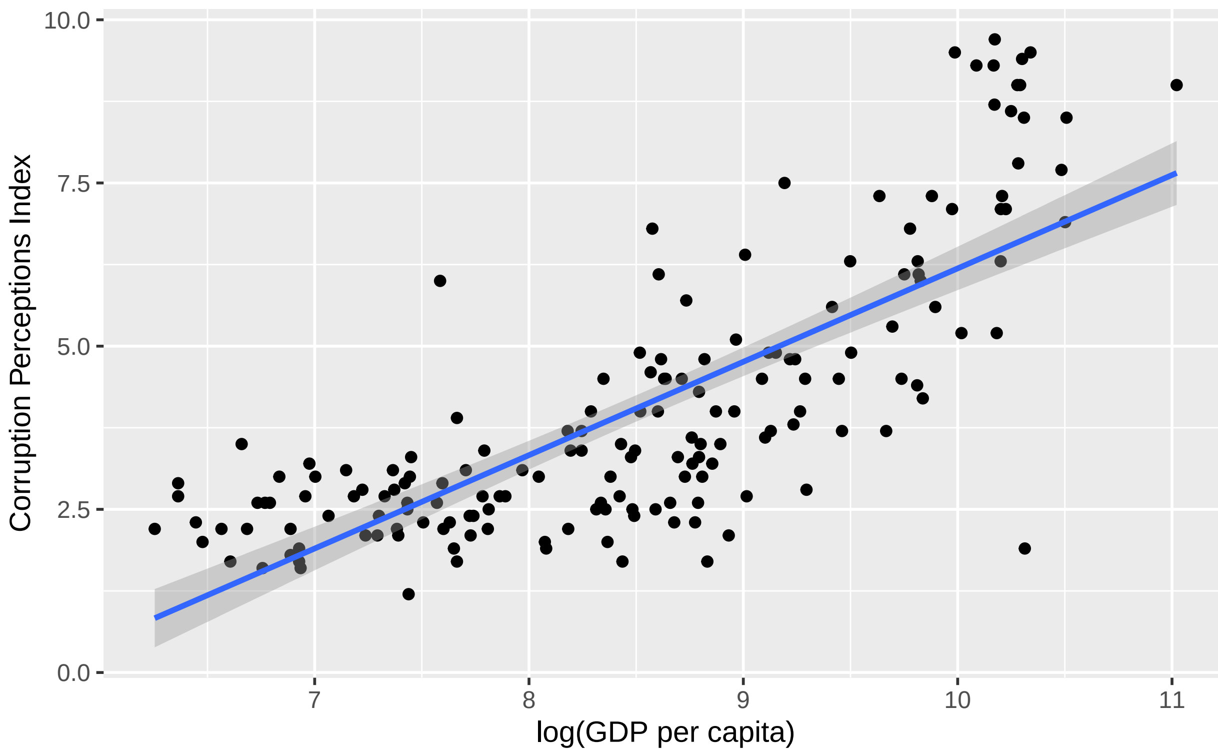
```
ggplot(df, aes(x = undp_gdp, y = ti_cpi)) +
  geom_point() +
  geom_smooth(method = "lm") +
  labs(x = "GDP per capita (PPP)", y = "Corruption Perceptions Index")
```



b) The relationship is positive—richer countries tend to be less corrupt—but the pattern is clearly non-linear. Most countries cluster at low GDP values, and the linear fit does not capture the curvature well.

c) Scatter plot with log-transformed GDP:

```
ggplot(df, aes(x = log(undp_gdp), y = ti_cpi)) +
  geom_point() +
  geom_smooth(method = "lm") +
  labs(x = "log(GDP per capita)", y = "Corruption Perceptions Index")
```



The log transformation spreads out the lower-income countries and compresses the upper tail, producing a much more linear relationship.

3. Bivariate regression

a–b) Estimate the level-level model:

```
m1 = lm(ti_cpi ~ undp_gdp, data = df)
tidy(m1)
```

```
## # A tibble: 2 x 5
##   term        estimate  std.error statistic  p.value
##   <chr>          <dbl>      <dbl>     <dbl>    <dbl>
## 1 (Intercept) 2.50     0.124           20.1 1.37e-46
## 2 undp_gdp    0.000173 0.00000929      18.6 1.12e-42
```

The coefficient on undp_gdp gives the predicted change in the corruption index for a one-dollar increase in GDP per capita. For a $10,000 increase, multiply the coefficient by 10,000:

```
coef(m1)["undp_gdp"] * 10000
```

```
## undp_gdp
## 1.729782
```

c) Predicted corruption at the 25th and 75th percentiles of GDP:

```
q25 = quantile(df$undp_gdp, 0.25)
q75 = quantile(df$undp_gdp, 0.75)
c(q25, q75)
```

```
##      25%      75%
```

```
##  1974.25 10862.50
```

```
predictions(m1, newdata = datagrid(undp_gdp = c(q25, q75)))
```

```
##
##  undp_gdp Estimate Std. Error    z Pr(>|z|)     S 2.5 % 97.5 %
##      1974     2.84     0.1130 25.2   <0.001 462.0  2.62   3.07
##     10862     4.38     0.0942 46.5   <0.001   Inf  4.20   4.57
##
## Columns: rowid, estimate, std.error, statistic, p.value, s.value, conf.low, conf.high, ti_cpi, undp_gdp
## Type:  response
```

The difference in predicted corruption between a country at the 75th percentile and one at the 25th percentile of GDP captures the interquartile range effect. The confidence intervals indicate the precision of these predictions.

4. Non-linear specifications

a–b) Log model:

```
m2 = lm(ti_cpi ~ log(undp_gdp), data = df)
tidy(m2)
```

```
## # A tibble: 2 x 5
##   term          estimate std.error statistic  p.value
##   <chr>            <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)      -8.11    0.769      -10.6 2.76e-20
## 2 log(undp_gdp)     1.43    0.0896      16.0 1.74e-35
```

In a level-log model, a 1% increase in GDP per capita is associated with a change of $\beta_1/100$ in the corruption index. For a doubling of GDP ($\log(2) \approx 0.693$):

```
coef(m2)["log(undp_gdp)"] * log(2)
```

```
## log(undp_gdp)
##     0.9915562
```

c) Quadratic model:

```
m3 = lm(ti_cpi ~ undp_gdp + I(undp_gdp^2), data = df)
tidy(m3)
```

```
## # A tibble: 3 x 5
##   term          estimate std.error statistic  p.value
##   <chr>            <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)    2.14e+0  1.42e- 1     15.1  4.71e-33
## 2 undp_gdp       2.63e-4  2.15e- 5     12.2  5.83e-25
## 3 I(undp_gdp^2) -2.49e-9  5.42e-10     -4.60 8.35e- 6
```

d) Compare R^2:

```
r2 = c(
  "Level-Level" = summary(m1)$r.squared,
  "Level-Log"   = summary(m2)$r.squared,
```

```r
  "Quadratic"    = summary(m3)$r.squared)
r2
```

```
## Level-Level   Level-Log   Quadratic
##   0.6734049   0.6025131   0.7101202
```

The log specification fits the data best, consistent with the scatter plots showing a concave relationship. A non-linear specification is appropriate because the marginal return to additional GDP diminishes at higher income levels: moving from $1,000 to $5,000 matters more for governance quality than moving from $25,000 to $29,000.

5. Marginal effects

a) Average marginal effect of GDP in the log model:

```r
avg_slopes(m2, variables = "undp_gdp")
```

```
##
##      Term Estimate Std. Error  z Pr(>|z|)     S    2.5 %   97.5 %
##  undp_gdp 0.000522  0.0000327 16   <0.001 188.0 0.000458 0.000587
##
## Columns: term, estimate, std.error, statistic, p.value, s.value, conf.low, conf.high
## Type:  response
```

b) The AME differs from the raw coefficient on `log(undp_gdp)` because the marginal effect of GDP in a level-log model depends on the level of GDP: $\partial y/\partial x = \beta/x$. The AME averages this over all observed values. It tells us the average predicted change in the corruption index for a one-dollar increase in GDP across all countries in the sample.

c) Marginal effects of the quadratic model at specific GDP values:

```r
slopes(m3, variables = "undp_gdp",
       newdata = datagrid(undp_gdp = c(2000, 10000, 30000)))
```

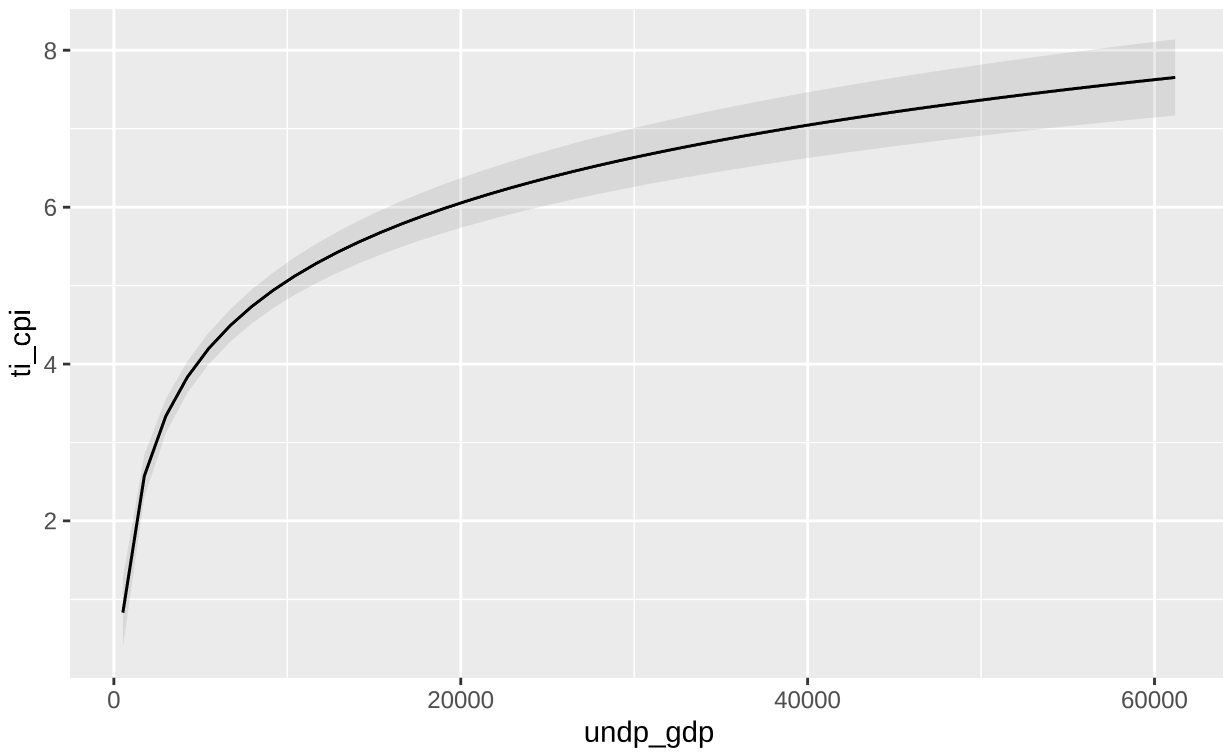
```
##
##      Term undp_gdp Estimate Std. Error     z Pr(>|z|)     S     2.5 %   97.5 %
##  undp_gdp     2000 0.000253  0.0000196 12.94   <0.001 124.9 0.0002151 0.000292
##  undp_gdp    10000 0.000214  0.0000125 17.15   <0.001 216.5 0.0001892 0.000238
##  undp_gdp    30000 0.000114  0.0000156  7.32   <0.001  41.9 0.0000834 0.000144
##
## Columns: rowid, term, estimate, std.error, statistic, p.value, s.value, conf.low, conf.high, undp_gdp, 
## Type:  response
```

The marginal effect of GDP on corruption diminishes as countries become richer. At low GDP levels, an additional dollar of income has a larger predicted effect on corruption than at high GDP levels. This is consistent with the concave shape of the relationship.

6. Prediction plots

a) Prediction plot for the log model:

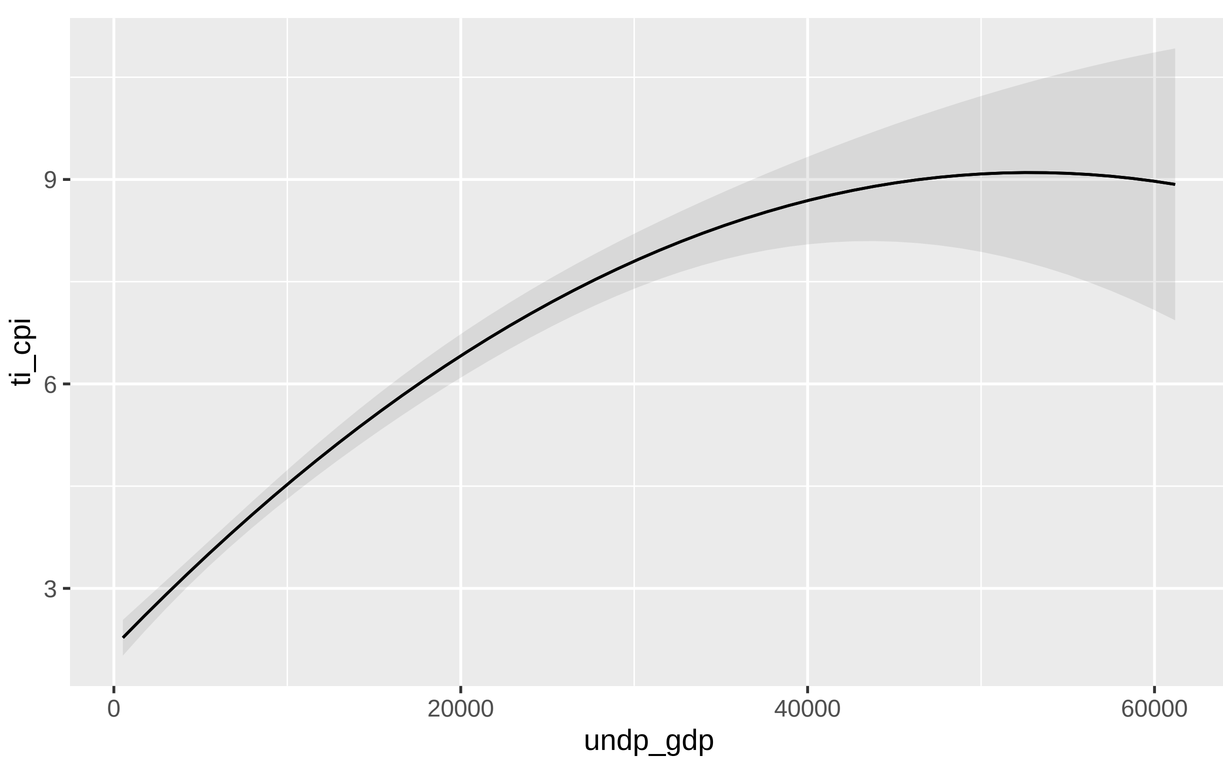
```r
p1 = plot_predictions(m2, condition = "undp_gdp")
p1
```




```r
ggsave("pred_plot_m2.png", p1, width = 6, height = 4)
```

b) Prediction plot for the quadratic model:

```r
p2 = plot_predictions(m3, condition = "undp_gdp")
p2
```



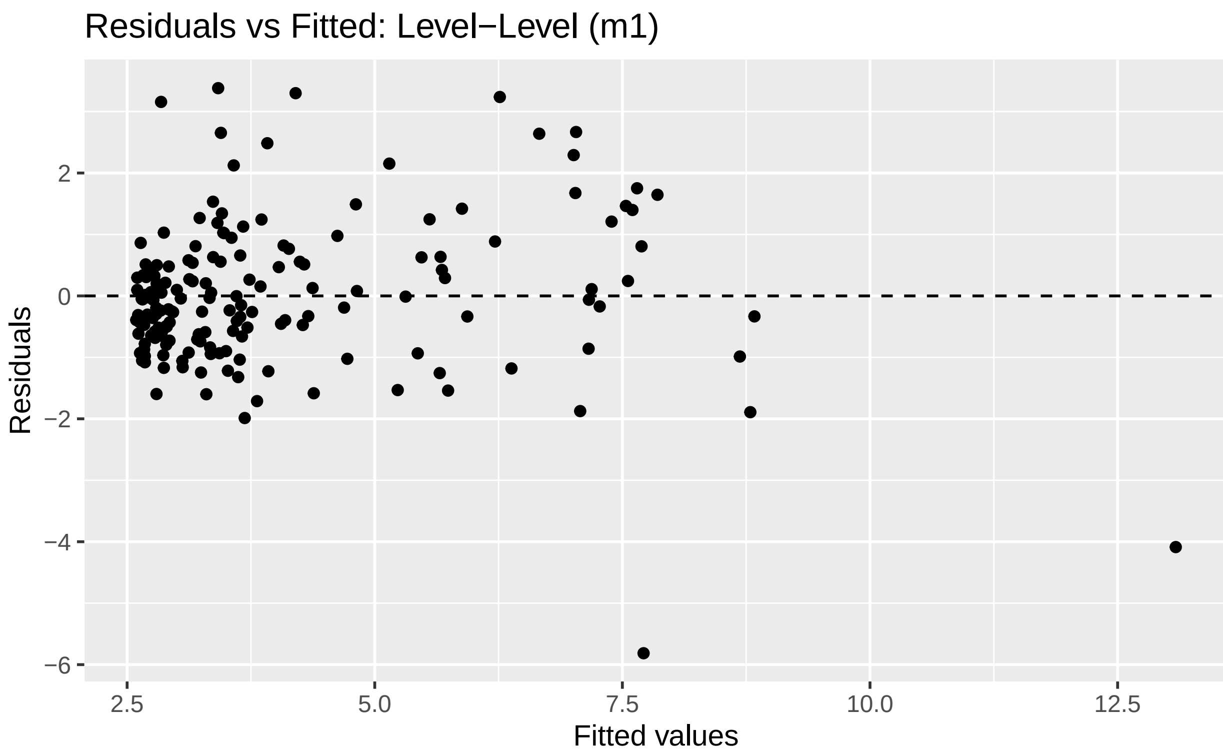
```
ggsave("pred_plot_m3.png", p2, width = 6, height = 4)
```

c) Both models tell a similar story: corruption decreases sharply with initial increases in GDP and then levels off at higher income levels. The log model produces a smoother curve, while the quadratic model can curve back upward at very high GDP values (a feature of the parabolic functional form that may not be substantively meaningful).

7. Residual diagnostics

a) Residuals vs. fitted for the level-level model:

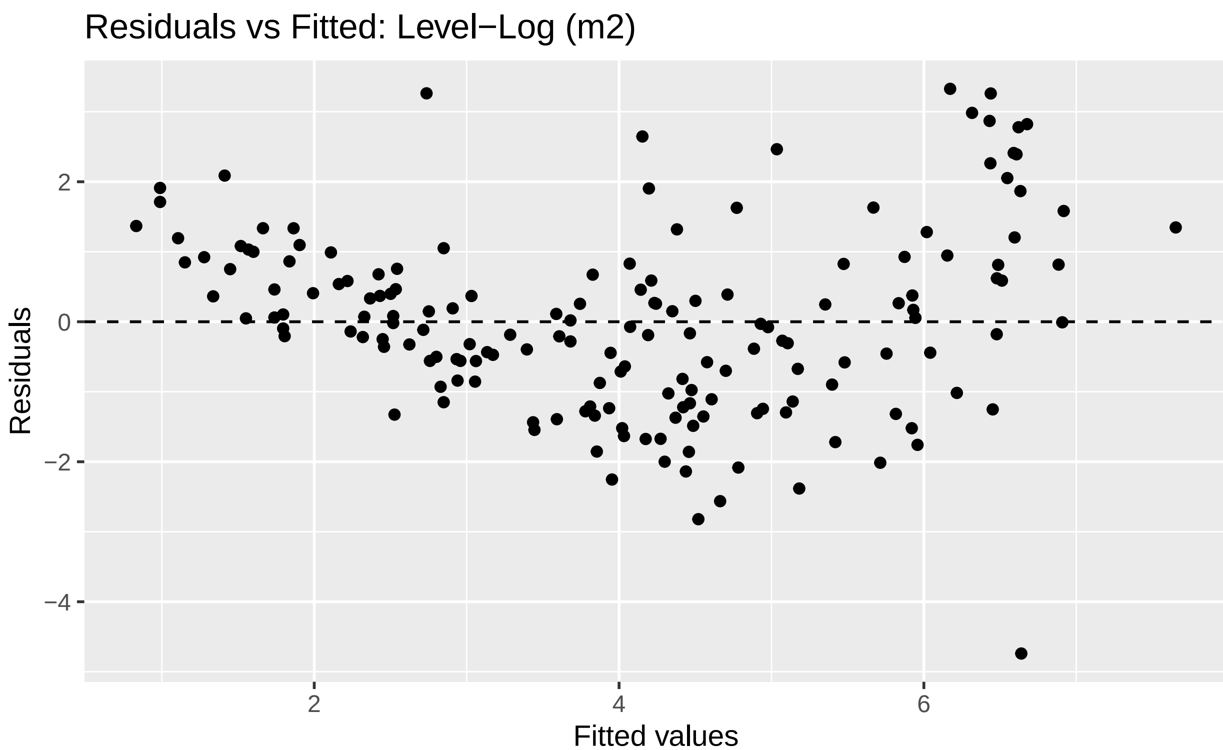
```
m1_aug = augment(m1)
ggplot(m1_aug, aes(x = .fitted, y = .resid)) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed") +
  labs(x = "Fitted values", y = "Residuals", title = "Residuals vs Fitted: Level-Level (m1)")
```



The residual plot shows a clear curved pattern, indicating that the linear specification misses the non-linear relationship. The spread of residuals also appears to increase with fitted values, suggesting heteroskedasticity.

b) Residuals vs. fitted for the log model:

```
m2_aug = augment(m2)
ggplot(m2_aug, aes(x = .fitted, y = .resid)) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed") +
  labs(x = "Fitted values", y = "Residuals", title = "Residuals vs Fitted: Level-Log (m2)")
```



The log transformation substantially improves the residual pattern. The curvature is reduced, though some heteroskedasticity may remain.

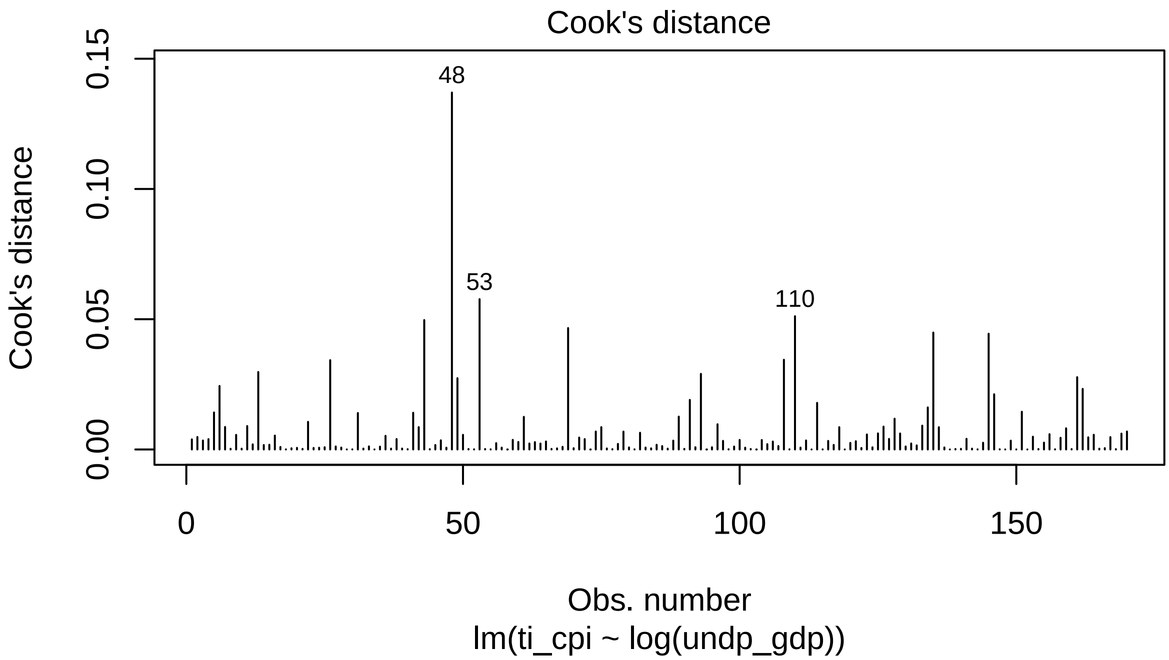
c) Cook's distance for influential observations:

```
n = nrow(df)
threshold = 4 / n

cooks_d = cooks.distance(m2)
influential = which(cooks_d > threshold)
df$cname[influential]
```

```
##  [1] "Australia"          "Bhutan"             "Canada"
##  [4] "Denmark"            "Equatorial Guinea" "Ethiopia"
##  [7] "Finland"            "Iceland"            "Malawi"
## [10] "Netherlands"        "New Zealand"        "Singapore"
## [13] "Sweden"             "United Kingdom"
```

```
plot(m2, which = 4)
```



d) Influential observations should not be removed automatically. They may represent genuine cases (e.g., very wealthy or very corrupt countries) rather than data errors. A recommended robustness check would be to re-estimate the model excluding these observations and compare the coefficients. If the results are similar, the original estimates are robust.

8. Publication-quality table

a) Regression table comparing all three models:

```
modelsummary(
  list("Level-Level" = m1, "Level-Log" = m2, "Quadratic" = m3),
  vcov = "robust",
  stars = TRUE,
  gof_map = c("r.squared", "nobs"),
  output = "markdown")
```

	Level-Level	Level-Log	Quadratic
(Intercept)	2.502***	-8.114***	2.139***
	(0.146)	(0.840)	(0.110)
undp_gdp	0.000***		0.000***
	(0.000)		(0.000)
log(undp_gdp)		1.431***	
		(0.104)	
I(undp_gdp^2)			0.000***
			(0.000)
R2	0.673	0.603	0.710
Num.Obs.	170	170	170

Note: + p < 0.1, * p < 0.05, ** p < 0.01, *** p < 0.001

b) The level-log model (m2) is the preferred specification. It has the highest R^2, produces the best residual diagnostics, and its functional form has a clear substantive interpretation: the relationship between wealth and corruption is one of diminishing returns. The log transformation also avoids the quadratic model's problem of an eventual sign reversal at extreme values.