

Introduction to Bivariate Linear Regression

Dave Clark

January 26, 2025








These slides introduce the basics of bivariate OLS regression using cross-sectional data. We'll analyze how infant mortality rates (IMR) vary across countries based on their political systems and economic development. Until the end, we'll focus on bivariate models. The goal is just to think through some of the elements of the model and questions we face when modeling - you'll benefit from having reviewed the matrix derivation slides, and the thinking about data slides.

Data Exploration

Let's examine the dataset's structure and summary statistics. The data include one observation for each country, and measures of infant mortality, GDP per capit and its natural log, deaths from natural disasters, whether there's systematic censorship, and the polity scale. Here are a few ways to look at the data.

```
tibble(censor)
```

```
# A tibble: 155 x 8
  ctry      ccode  IMR  gdppc deaths censorship polity lngdp
  <chr>    <int> <dbl> <dbl> <int>      <int> <int> <dbl>
1 Afghanistan  700 177.  1272.  7308         1    -7  7.15
2 Albania      339  45.7  2833.   125         1     7  7.95
3 Algeria      615  56.5  4538.   921         1    -2  8.42
4 Angola       540 192.  1164.   766         1    -1  7.06
5 Argentina    160  24.2  8616.   112         1     8  9.06
6 Armenia      371  26.7  3231.     4         1     5  8.08
7 Australia    900   7.18 18623.    75         1    10  9.83
8 Austria      305   7.84 19533.    53         1    10  9.88
9 Azerbaijan   373  81.5  2955.    42         1    -6  7.99
10 Bahrain     692  31.8 11702.     0         1    -9  9.37
# i 145 more rows
```

	Unique	Missing Pct.	Mean	SD	Min	Median	Max	Histogram
ccode	154	0	461.6	234.9	2.0	452.0	950.0	
IMR	154	0	55.9	44.4	3.4	45.9	191.8	
gdppc	154	0	6050.3	6092.3	371.1	3667.9	26653.5	
deaths	130	0	6499.1	30349.9	0.0	180.0	300000.0	
censorship	2	0	0.9	0.2	0.0	1.0	1.0	
polity	21	1	4.3	6.2	-10.0	7.0	10.0	
lngdp	154	0	8.2	1.1	5.9	8.2	10.2	

```
datasummary_skim(censor)
```

```
censor %>%
  summarise_if(is.numeric, median, na.rm=TRUE)
```

```
   ccode   IMR   gdppc deaths censorship polity   lngdp
1   452 45.865 3667.933   180          1     7 8.207383
```

```
ggpairs(censor, columns = 3:7)
```

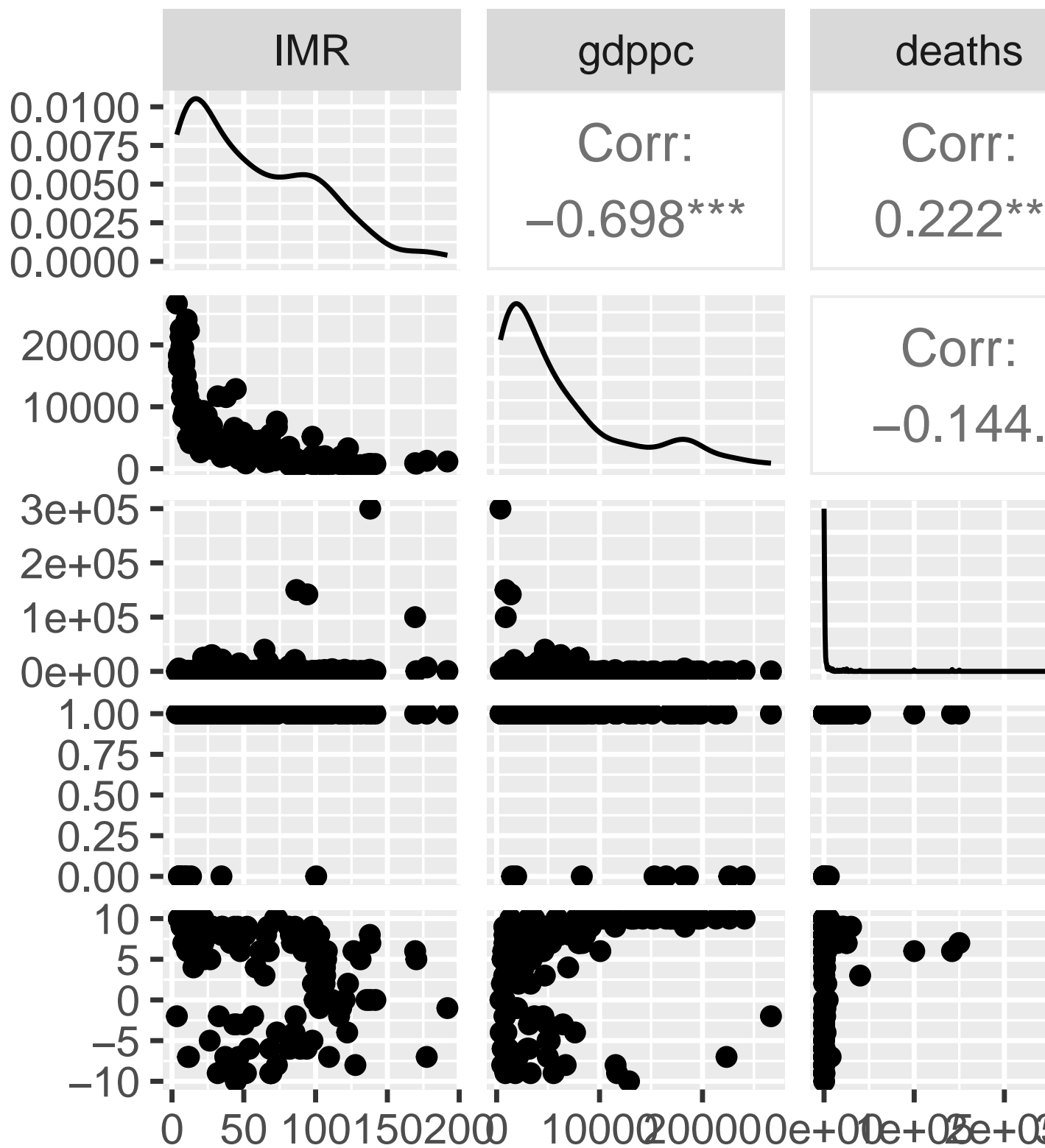


Table 1: Null Model

(1)	
(Intercept)	55.946***
	(3.566)
Num.Obs.	155
+ p < 0.1, * p < 0.05, ** p < 0.01, *** p < 0.001	

Constant-Only Model

Let's start with the constant-only, or null model:

```
m0 <- lm(IMR ~ 1, data = censor)

modelsummary(
  models = m0,
  stars = TRUE,
  title = "Null Model",
  gof_omit = 'AIC|Log.Lik|R2|BIC|RMSE',
  #gof_map = c("nobs", "r.squared", "adj.r.squared", "f.statistic", "p.value", "sigma")
)
```

The estimate of β_0 is the mean of the dependent variable, IMR. The constant-only model is a simple way to understand the data's central tendency.

```
summary(censor$IMR)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 3.388 14.934  45.865  55.946  90.438 191.787
```

Political System and Infant Mortality

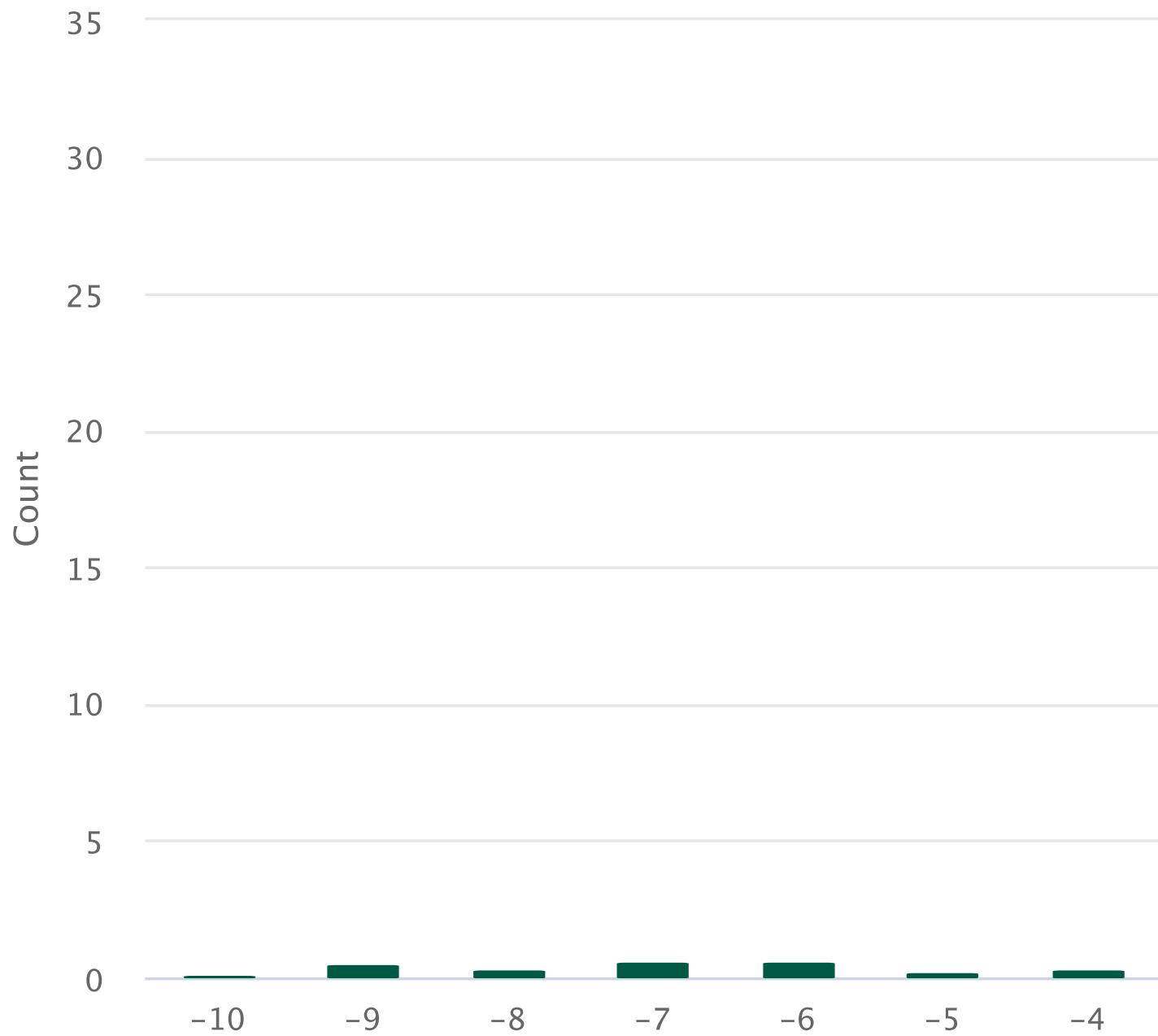
Now we'll examine how a country's political system (measured by the Polity score) affects infant mortality. The Polity score ranges from -10 (most autocratic) to +10 (most democratic). Let's look at the Polity variable:

```

# Calculate counts by polity score
polity_counts <- table(censor$polity)
polity_df <- data.frame(
  polity = as.numeric(names(polity_counts)),
  count = as.numeric(polity_counts)
)

# Create highchart
highchart() %>%
  hc_chart(type = "column") %>%
  hc_xAxis(categories = polity_df$polity, title = list(text = "Polity Score")) %>%
  hc_yAxis(title = list(text = "Count")) %>%
  hc_add_series(
    data = polity_df$count,
    name = "Countries",
    color = "#005A43",
    dataLabels = list(
      enabled = TRUE,
      format = "{y}"
    )
  ) %>%
  hc_title(text = "Distribution of Polity Scores") %>%
  hc_plotOptions(column = list(
    borderRadius = 3,
    borderWidth = 0
  ))

```



```
#  
#
```

Polity is categorical and ordered. It's commonly (mis)treated as a continuous variable in the literature - we'll start there:

```
m1 <- lm(IMR ~ polity, data=censor)  
  
summary(m1)
```

Call:

```
lm(formula = IMR ~ polity, data = censor)
```

Residuals:

Min	1Q	Median	3Q	Max
-76.97	-31.95	-11.67	35.30	120.50

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	68.4898	4.0820	16.778	< 2e-16 ***
polity	-2.7999	0.5431	-5.155	7.84e-07 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 41.22 on 151 degrees of freedom
(2 observations deleted due to missingness)

Multiple R-squared: 0.1496, Adjusted R-squared: 0.144

F-statistic: 26.57 on 1 and 151 DF, p-value: 7.836e-07

The estimated coefficient of -2.7999 suggests that, on average, each unit increase in the Polity score is associated with a 2.7999 decrease in the IMR; the estimate is statistically different from zero.

Let's take a brief detour to think about the errors in our model. First, let's compute the sum of squared residuals, σ^2 , and the residual standard error. Remember the residuals are given by $y - \hat{y}$; σ^2 is the sum of squared residuals divided by $n - k - 1$; and the residual standard error is the square root of σ^2 .

```
sse <- sum(residuals(m1)^2)
sigma2 <- sse / m1$df.residual
rse <- sqrt(sigma2)

cat("Sum of Squared Residuals:", sse, "\n")
```

Sum of Squared Residuals: 256597.8

```
cat("Sigma^2:", sigma2, "\n")
```

Sigma^2: 1699.323

```
cat("Residual Standard Error:", rse, "\n")
```

Residual Standard Error: 41.22285

Comparing this to the estimates above, you'll see our estimate of the RSE is the same as R's estimate. This is the average residual in the model in terms of the y variable, IMR - so on average, we're off by about 41 infant deaths per 1000 live births.

Generate predictions

Let's compute \hat{y} for each category of Polity and plot along with the actual observed value of IMR for each country.

```
predictions <- data.frame(polity=numeric(0), xb=numeric(0))

for (i in seq(1,21,1)){
  xb = 68.4898 + -2.7999*(i-11)
  predictions[i:i,] <- data.frame(i-11, xb)
}

# # Visualize predictions with actual data
# ggplot(data=censor, aes(x=polity, y=IMR)) +
#   geom_point(color="green") +
#   geom_text(label=censor$ctry, size=3) +
#   geom_line(data=predictions, aes(x=polity, y=xb)) +
#   labs(x="Polity", y="Infant Mortality Rate")
```



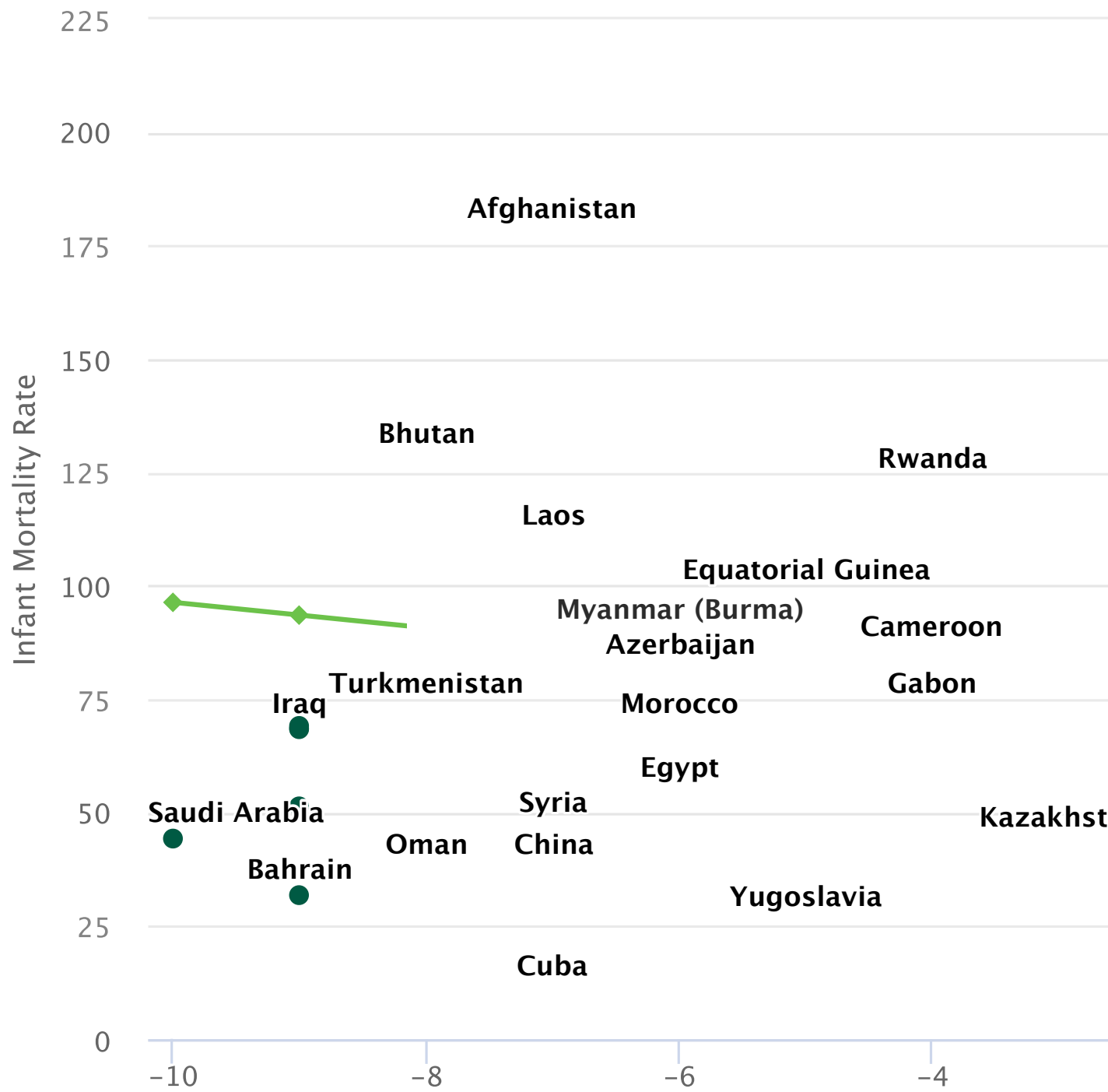
```

library(highcharter)

# Generate predictions dataframe
predictions <- data.frame(
  polity = seq(-10, 10, 1),
  xb = 68.4898 + -2.7999 * (seq(1, 21, 1) - 11)
)

highchart() %>%
  hc_add_series(
    data = censor,
    name = "Observed IMR",
    type = "scatter",
    hcaes(x = polity, y = IMR),
    color = "#005A43",
    dataLabels = list(enabled = TRUE, format = "{point.ctr}")
  ) %>%
  hc_add_series(
    data = predictions,
    name = "Predicted IMR",
    type = "line",
    color="#6CC24A",
    hcaes(x = polity, y = xb)
  ) %>%
  hc_xAxis(title = list(text = "Polity")) %>%
  hc_yAxis(title = list(text = "Infant Mortality Rate"))

```



Rethinking Polity

Since Polity is categorical, we don't know the intervals between categories, so interpreting this coefficient (-2.79) as an effect of a 1 unit change on y really doesn't make much sense. We are assuming the effect of a change in Polity is the same across all unit changes in the Polity scale - not only are we assuming the change is the same magnitude, but that it's in the same direction - that the effect is linear.

Another way to treat Polity in this model is to include it as a factor variable. This allows us to estimate the effect of each level of Polity on IMR, relative to a reference level. In effect, we are including a dummy variable for all but one of the categories of Polity. Let's look at such a model:

```
m1f <- lm(IMR ~ factor(polity), data=censor)

modelsummary(
  models = m1f,
  stars = TRUE,
  title = "OLS Estimates",
  gof_omit = 'AIC|Log.Lik|R2|BIC|RMSE',
  #gof_map = c("nobs", "r.squared", "adj.r.squared", "f.statistic", "p.value", "sigma")
)
```

Let's generate predictions from the factor model. Each bar represents the 95% confidence interval for the predicted IMR at each level of Polity; the dots are the predictions.

```
# Predictions for factor model
new_data <- data.frame(polity = factor(seq(-10, 10),
                                     levels = sort(unique(censor$polity))))
predictions <- predict(m1f, newdata = new_data, interval = "confidence")
results <- data.frame(
  polity = as.numeric(as.character(new_data$polity)),
  fit = predictions[, "fit"],
  lwr = predictions[, "lwr"],
  upr = predictions[, "upr"]
)

results <- results[!is.na(results$fit), ]
results <- results[order(results$polity), ]

highchart() %>%
  hc_chart(type = "errorbar") %>%
  hc_xAxis(
```

Table 2: OLS Estimates

	(1)
(Intercept)	44.411 (34.102)
factor(polity)-9	13.518 (37.356)
factor(polity)-8	35.063 (39.377)
factor(polity)-7	21.136 (36.834)
factor(polity)-6	33.683 (36.834)
factor(polity)-5	17.644 (41.766)
factor(polity)-4	49.030 (39.377)
factor(polity)-3	1.577 (39.377)
factor(polity)-2	14.589 (37.356)
factor(polity)-1	93.026* (39.377)
factor(polity)0	79.959* (36.834)
factor(polity)2	64.582 (39.377)
factor(polity)3	41.135 (41.766)
factor(polity)4	25.645 (38.127)
factor(polity)5	49.484 (36.834)
factor(polity)6	35.730 (35.494)
factor(polity)7	7.281 (35.389)
12	
factor(polity)8	8.292 (34.987)
factor(polity)9	-2.346 (35.036)
factor(polity)10	-30.601

```

    categories = results$polity,
    title = list(text = "Polity Score")
) %>%
hc_yAxis(
  title = list(text = "Predicted IMR")
) %>%
hc_add_series(
  type = "errorbar",
  data = list_parse(
    data.frame(
      low = results$lwr,
      high = results$upr
    )
  ),
  name = "95% CI",
  color = "#005A43",
  stemWidth = 3,
  whiskerLength = 10
) %>%
hc_add_series(
  type = "scatter",
  data = results$fit,
  name = "Predicted IMR",
  color = "#005A43",
  marker = list(
    symbol = "circle",
    radius = 6
  ),
  dataLabels = list(
    enabled = TRUE,
    format = "{point.y:.1f}",
    verticalAlign = "bottom",
    y = -10
  )
) %>%
hc_title(text = "Predicted Infant Mortality Rates by Polity Score")

```

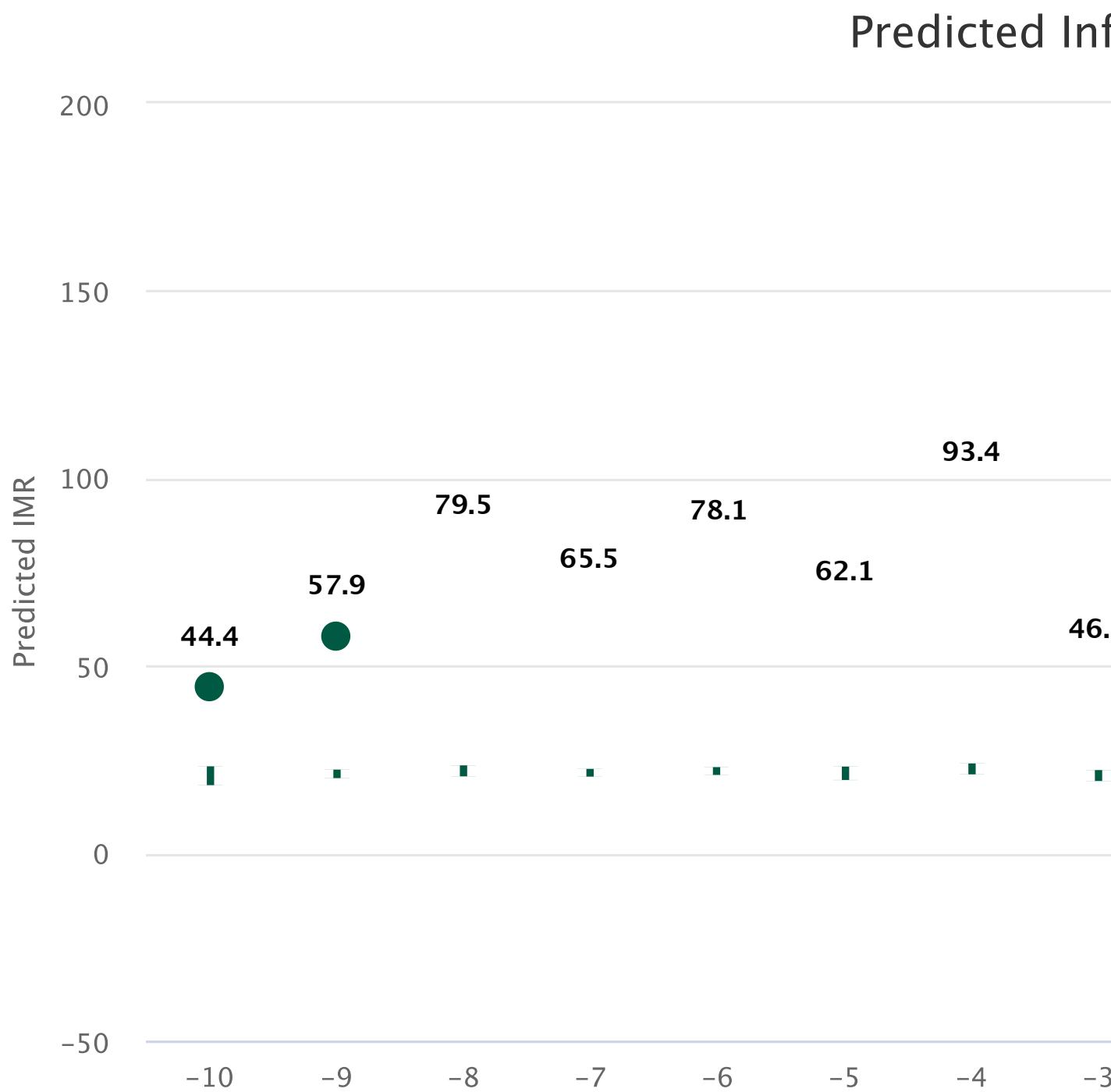


Table 3: OLS Estimates	
	(1)
(Intercept)	86.710*** (3.618)
gdppc	-0.005*** (0.000)
Num.Obs.	155
F	145.103
+ p < 0.1, * p < 0.05, ** p < 0.01, *** p < 0.001	

A couple of things to note regarding these predictions. Because Polity is categorical, we cannot draw a continuous line across the categories. You'll notice that it appears from the coefficients and from the predictions that the effect of Polity on IMR is not linear. The common practice of treating Polity as a continuous variable would be problematic in this model (and in most). Also, notice the ranges of the confidence intervals vary quite a lot. The variability within Polity category raises potentially interesting questions about development and regime.

Economic Development and Infant Mortality

Let's examine how GDP per capita affects infant mortality rates, exploring different functional forms.

Linear Relationship

```
m2 <- lm(IMR ~ gdppc, data=censor)

modelsummary(
  models = m2,
  stars = TRUE,
  title = "OLS Estimates",
  gof_omit = 'AIC|Log.Lik|R2|BIC|RMSE',
  #gof_map = c("nobs", "r.squared", "adj.r.squared", "f.statistic", "p.value", "sigma")
)
```

Looking at the predictions, it's pretty clear the scatterplot of IMR and GDP per capita doesn't suggest a linear relationship.

```

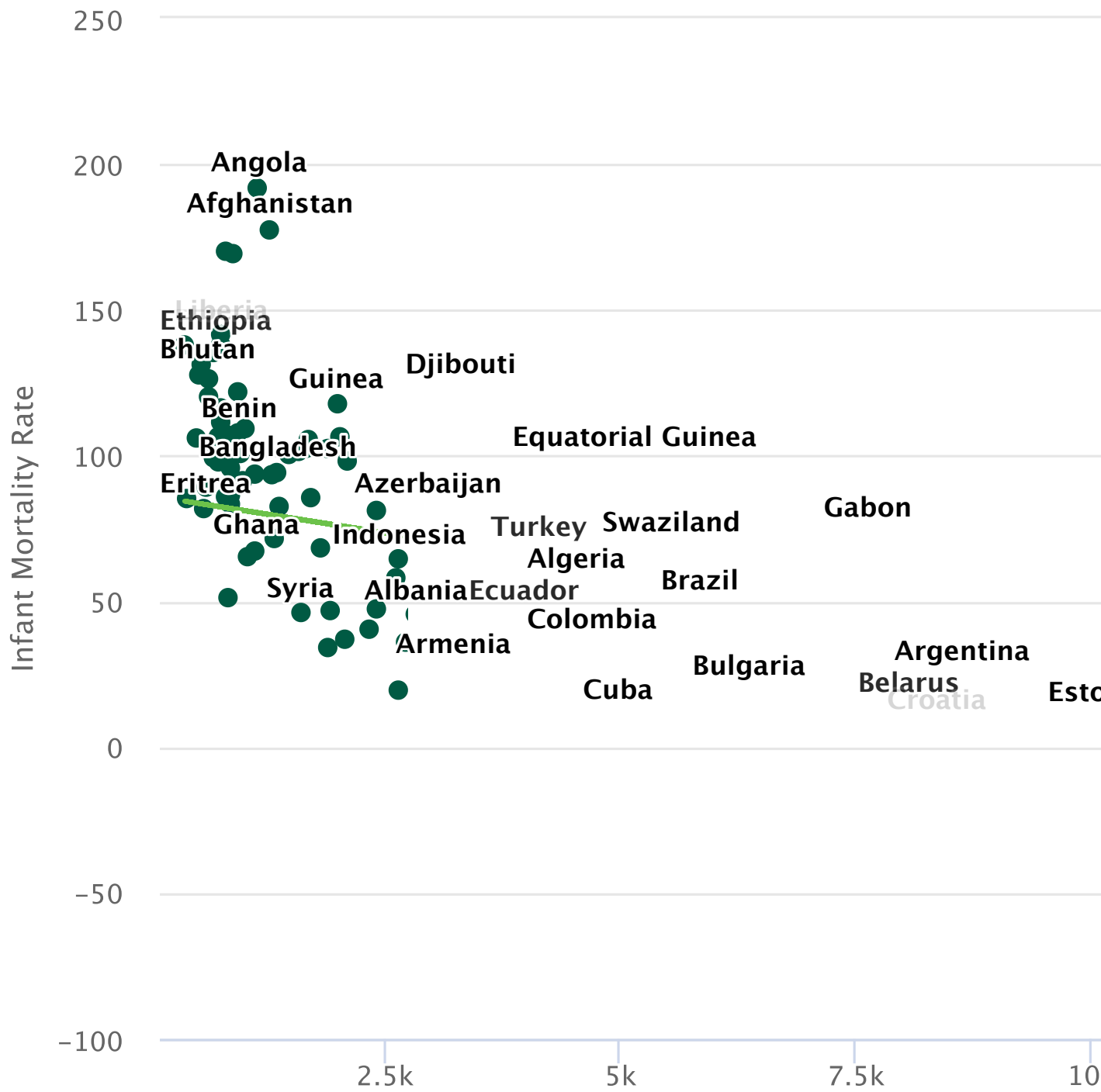
# Generate predictions
analysisdata <- censor %>%
  mutate(xb = coef(m2)[1] + coef(m2)[2]*gdppc)

pred <- predict(m2, interval="confidence", se.fit=TRUE)
analysisdata <- cbind(analysisdata, pred)
#
# # Plot with predictions
# ggplot(analysisdata, aes(x=gdppc, y=IMR)) +
#   geom_point(color="green") +
#   geom_text(label=censor$ctry, size=2) +
#   geom_line(aes(x=gdppc, y=xb)) +
#   labs(x="GDP per capita", y="Infant Mortality Rate")

# same as above using highcharter

highchart() %>%
  hc_add_series(
    data = censor,
    name = "Observed IMR",
    type = "scatter",
    hcaes(x = gdppc, y = IMR),
    color = "#005A43",
    dataLabels = list(enabled = TRUE, format = "{point.ctry}")
  ) %>%
  hc_add_series(
    data = analysisdata,
    name = "Predicted IMR",
    type = "line",
    color="#6CC24A",
    hcaes(x = gdppc, y = xb)
  ) %>%
  hc_xAxis(title = list(text = "GDP per capita")) %>%
  hc_yAxis(title = list(text = "Infant Mortality Rate"))

```

Residuals

Let's examine the residuals from this model - recall the residuals are the differences between the observed and predicted values of y .

```
# Calculate residuals
analysisdata <- analysisdata %>%
  mutate(res = IMR - xb)
#
# ggplot(analysisdata, aes(x=gdppc, y=res)) +
#   geom_point(color="green") +
#   geom_text(label=censor$ctry, size=3) +
#   geom_abline(slope=0, intercept=0) +
#   labs(x="GDP per capita", y="Residuals")

# same plot in highcharter

highchart() %>%
  hc_add_series(
    data = analysisdata,
    name = "Residuals",
    type = "scatter",
    hcaes(x = gdppc, y = res),
    color = "#005A43",
    dataLabels = list(enabled = TRUE, format = "{point.ctry}")
  ) %>%
  hc_add_series(
    data = list(list(x = min(analysisdata$gdppc), y = 0),
                list(x = max(analysisdata$gdppc), y = 0)),
    type = "line",
    color = "red",
    enableMouseTracking = FALSE,
    showInLegend = FALSE
  ) %>%
  hc_xAxis(title = list(text = "GDP per capita")) %>%
  hc_yAxis(title = list(text = "Residuals"))
```

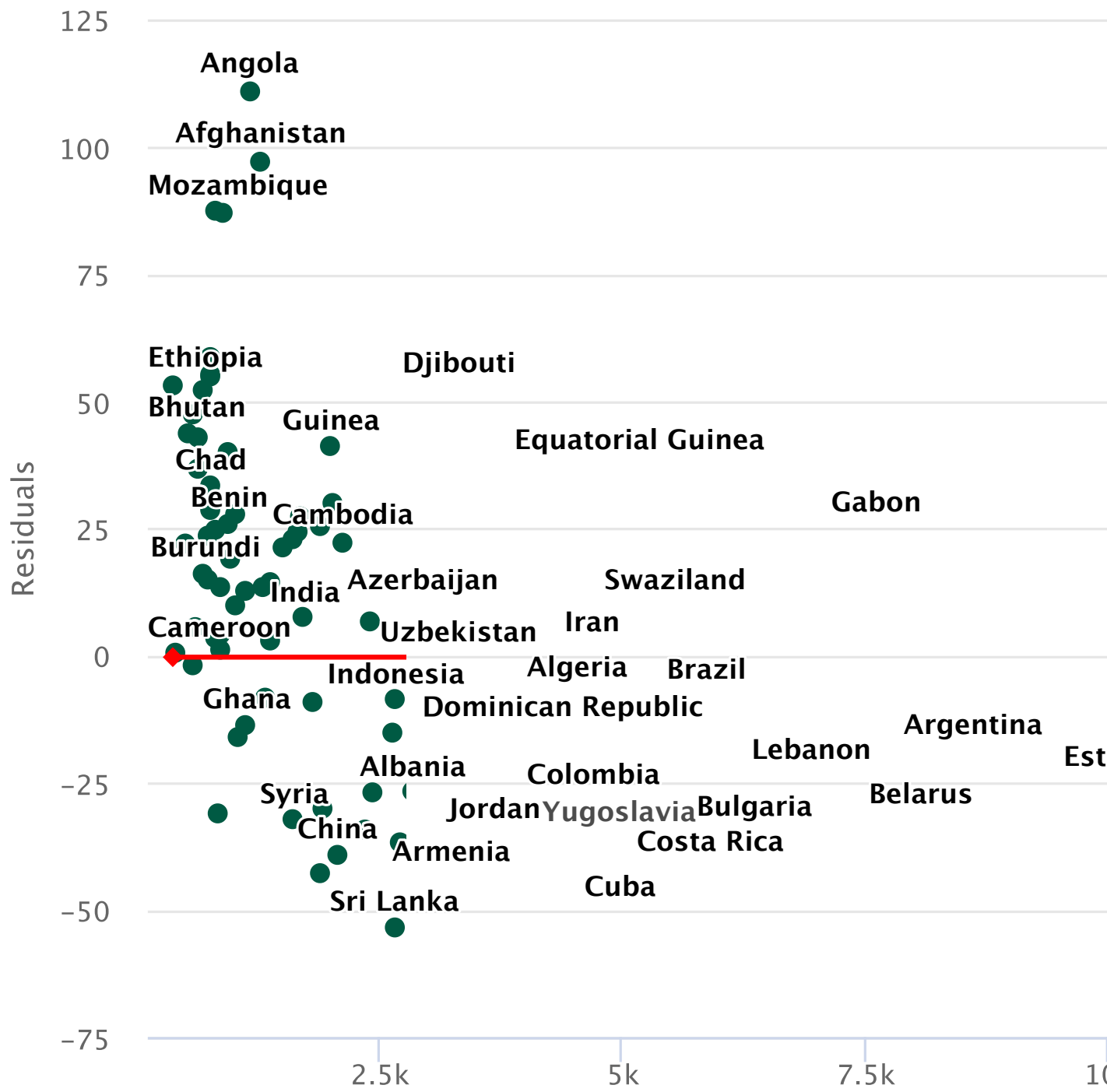


Table 4: OLS Estimates	
	(1)
(Intercept)	332.552*** (14.657)
lngdp	-33.846*** (1.777)
Num.Obs.	155
F	362.586
+ p < 0.1, * p < 0.05, ** p < 0.01, *** p < 0.001	

Both the predictions and the residuals suggest the relationship between GDP and IMR is not linear. Let's consider some alternatives that might fit the data better.

Log-transformed GDP

Let's try a log transformation of GDP per capita $\ln(GDP_{pc})$ and see if that improves the model.

```
analysisdata2 <- censor %>%
  mutate(lngdp = log(gdppc))

m3 <- lm(IMR ~ lngdp, data=analysisdata2)

modelsummary(
  models = m3,
  stars = TRUE,
  title = "OLS Estimates",
  gof_omit = 'AIC|Log.Lik|R2|BIC|RMSE',
  #gof_map = c("nobs", "r.squared", "adj.r.squared", "f.statistic", "p.value", "sigma")
)
```

and take a look at the predictions from this model:

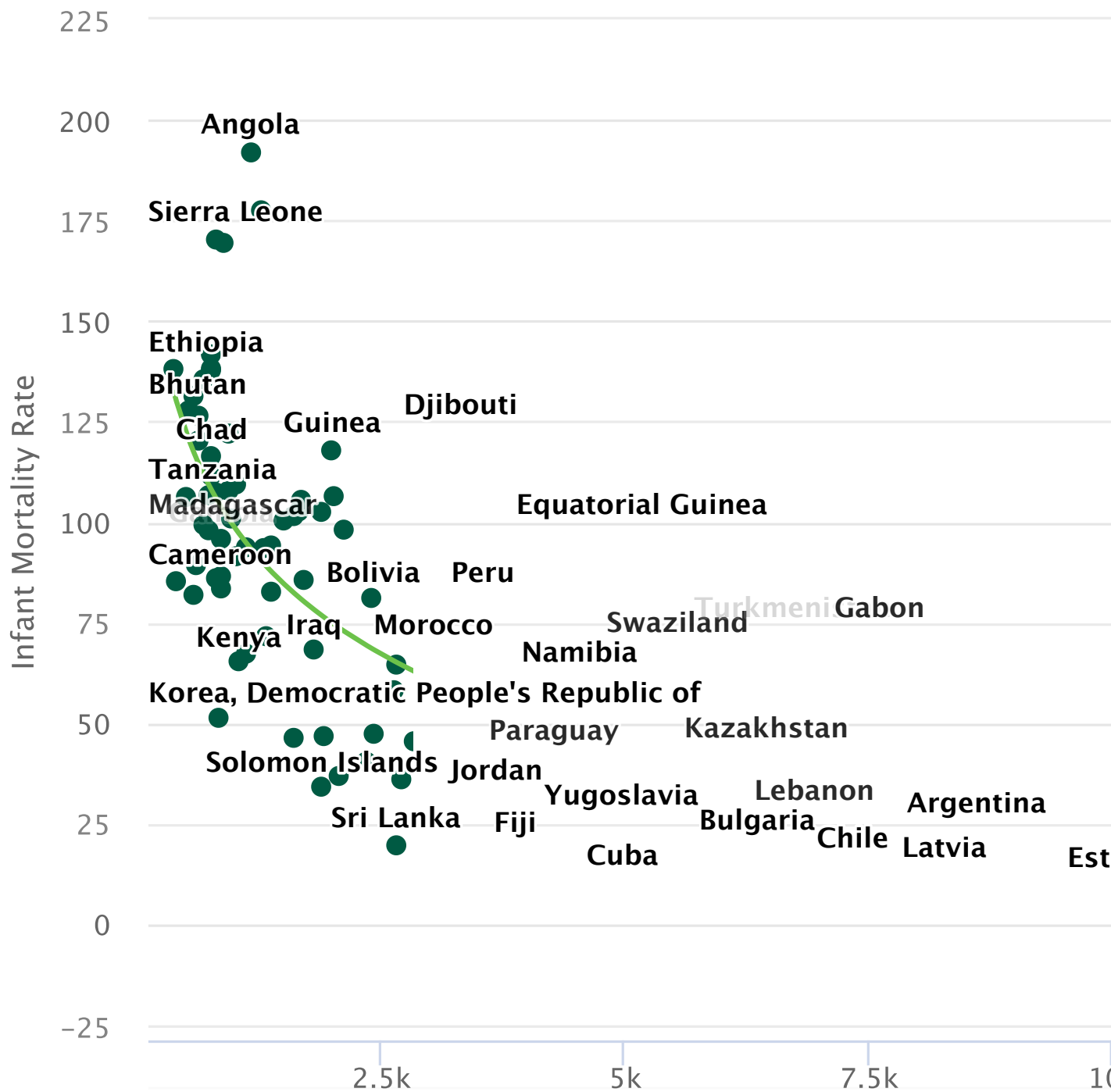
```
# Generate predictions
analysisdata2 <- analysisdata2 %>%
  mutate(xb = coef(m3)[1] + coef(m3)[2]*lngdp)
#sort by lngdp
analysisdata2 <- analysisdata2[order(analysisdata2$lngdp),]
```

```

# plot using highcharter

highchart() %>%
  hc_add_series(
    data = analysisdata2,
    name = "Observed IMR",
    type = "scatter",
    hcaes(x = gdppc, y = IMR),
    color = "#005A43",
    dataLabels = list(enabled = TRUE, format = "{point.ctr}")
  ) %>%
  hc_add_series(
    data = analysisdata2,
    name = "Predicted IMR",
    type = "line",
    color="#6CC24A",
    hcaes(x = gdppc, y = xb)
  ) %>%
  hc_xAxis(title = list(text = "GDP per capita")) %>%
  hc_yAxis(title = list(text = "Infant Mortality Rate"))

```



The log-transformed model seems to fit the data better than the linear model. The residuals are also more evenly distributed around zero, though still appear correlated with GDP per capita.

```
# Calculate residuals
analysisdata2 <- analysisdata2 %>%
  mutate(res = IMR - xb)

# Plot using highcharter

highchart() %>%
  hc_add_series(
    data = analysisdata2,
    name = "Residuals",
    type = "scatter",
    hcaes(x = gdppc, y = res),
    color = "#005A43",
    dataLabels = list(enabled = TRUE, format = "{point.ctr}")
  ) %>%
  hc_add_series(
    data = list(list(x = min(analysisdata2$gdppc), y = 0),
                 list(x = max(analysisdata2$gdppc), y = 0)),
    type = "line",
    color = "red",
    enableMouseTracking = FALSE,
    showInLegend = FALSE
  ) %>%
  hc_xAxis(title = list(text = "GDP per capita")) %>%
  hc_yAxis(title = list(text = "Residuals"))
```

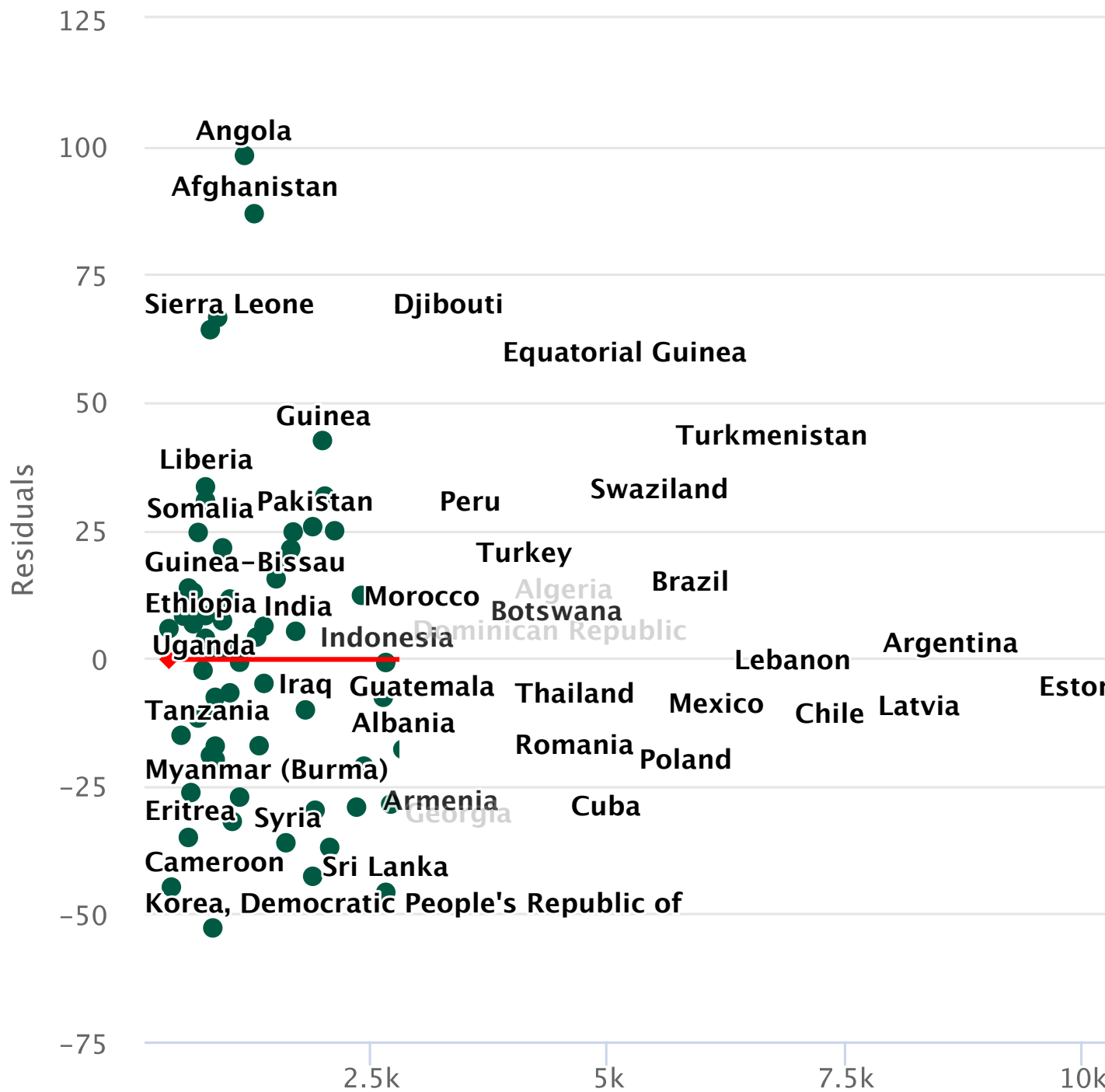


Table 5: OLS Estimates	
	(1)
(Intercept)	110.123*** (4.016)
gdppc	-0.015*** (0.001)
gdp2	0.000*** (0.000)
Num.Obs.	155
F	145.206
+ p < 0.1, * p < 0.05, ** p < 0.01, *** p < 0.001	

Polynomials

Finally, let's try a polynomial model to see if we can improve the fit further. We'll try a quadratic model, $IMR = \beta_0 + \beta_1 \times GDP_{pc} + \beta_2 \times GDP_{pc}^2$.

```

sensor$gdp2 = sensor$gdppc^2

m4 <- lm(IMR ~ gdppc + gdp2, data=sensor, na.action = na.omit)

modelsummary(
  models = m4,
  stars = TRUE,
  title = "OLS Estimates",
  gof_omit = 'AIC|Log.Lik|R2|BIC|RMSE',
)

```

GDP per capita is negatively related to IMR, then positively related at higher values. Here are predictions from the polynomial model (including a 95% confidence interval) and the actual data points:

```

fit2 <- predict(m4, interval="confidence", se.fit=TRUE)

predictions2 <- cbind(sensor, fit2)

# sort predictions2 by gdppc
predictions2 <- predictions2[order(predictions2$gdppc),]

```

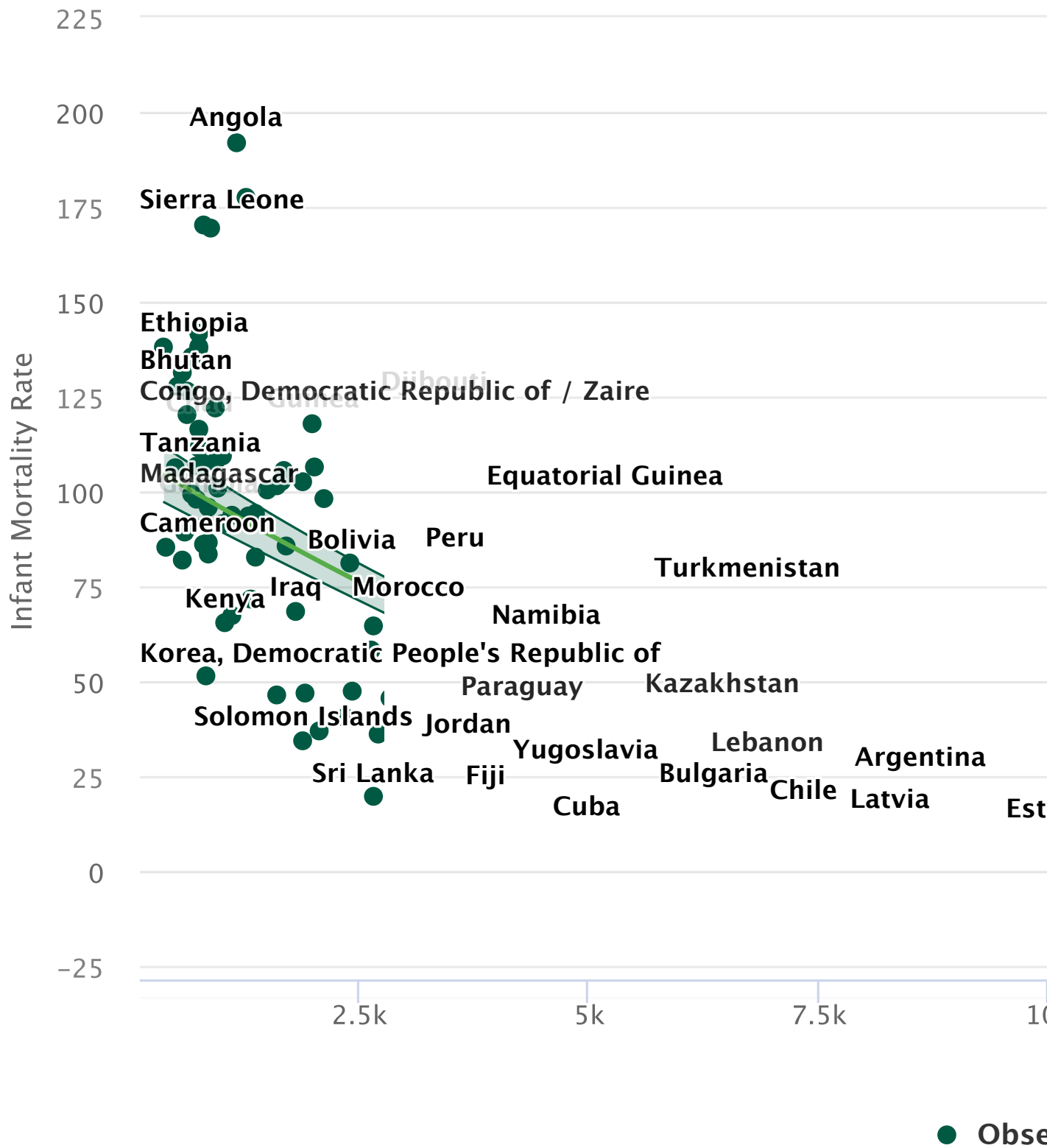
```

# ggplot(data=predictions2, aes(x=gdppc, y=IMR))+
#   geom_point(color="green") +   geom_text(label=predictions2$ctry, size=2) +
#   geom_line(aes(x=gdppc, y=fit.fit)) +
#   geom_ribbon(aes(ymin = fit.lwr, ymax = fit.upr), alpha = 0.2)

#plot using highcharter

highchart() %>%
  hc_add_series(
    data = predictions2,
    name = "Observed IMR",
    type = "scatter",
    hcaes(x = gdppc, y = IMR),
    color = "#005A43",
    dataLabels = list(enabled = TRUE, format = "{point.ctry}")
  ) %>%
  hc_add_series(
    data = predictions2,
    name = "Predicted IMR",
    type = "line",
    color="#6CC24A",
    hcaes(x = gdppc, y = fit.fit)
  ) %>%
  hc_add_series(
    data = predictions2,
    type = "arearange",
    name = "95% CI",
    hcaes(x = gdppc, low = fit.lwr, high = fit.upr),
    color = "#005A43",
    fillOpacity = 0.2
  ) %>%
  hc_xAxis(title = list(text = "GDP per capita")) %>%
  hc_yAxis(title = list(text = "Infant Mortality Rate"))

```



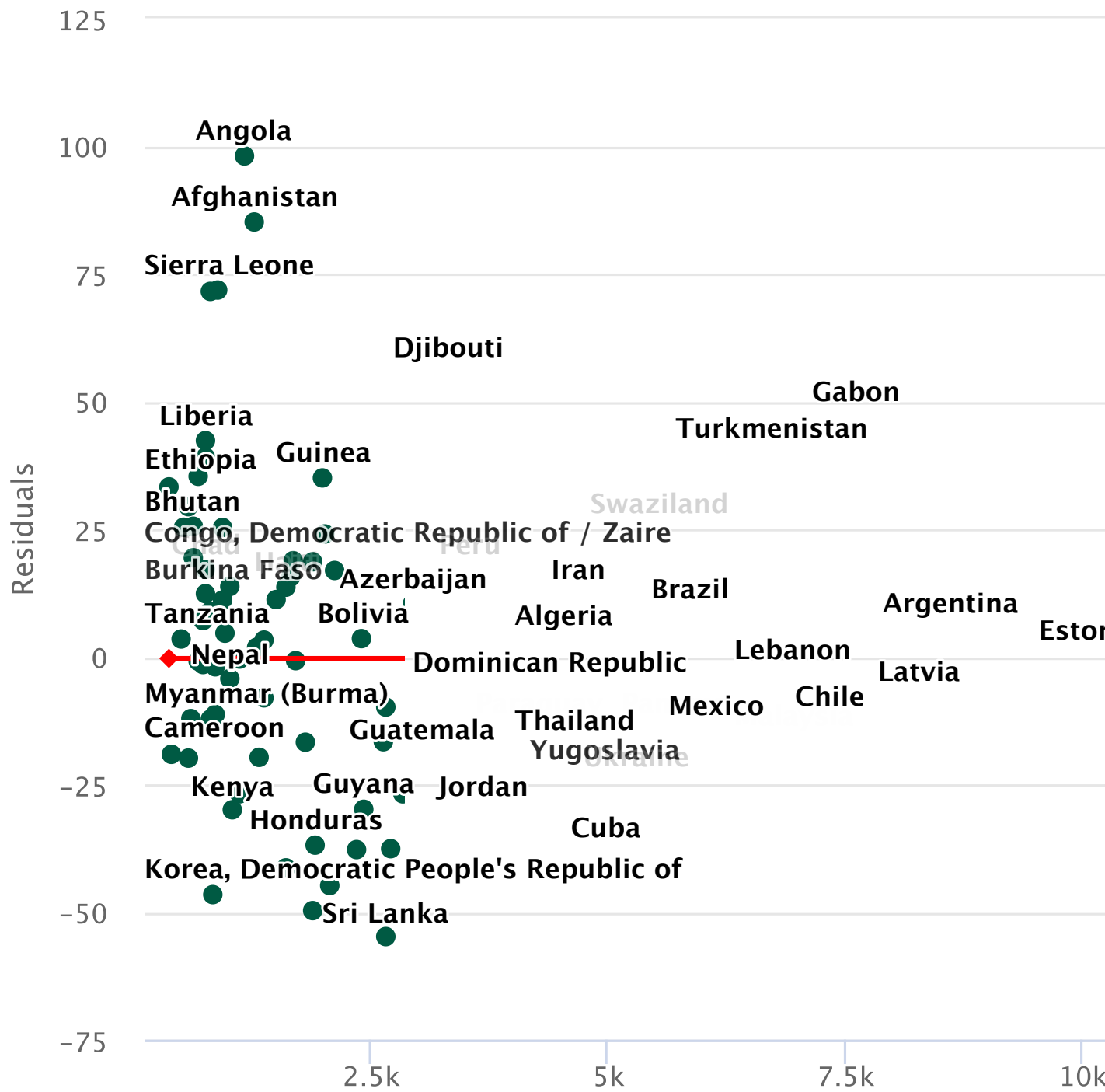
And let's look at the residuals from the polynomial model:

```
# Calculate residuals

predictions2 <- predictions2 %>%
  mutate(res = IMR - fit.fit)

# Plot using highcharter

highchart() %>%
  hc_add_series(
    data = predictions2,
    name = "Residuals",
    type = "scatter",
    hcaes(x = gdppc, y = res),
    color = "#005A43",
    dataLabels = list(enabled = TRUE, format = "{point.ctr}")
  ) %>%
  hc_add_series(
    data = list(list(x = min(predictions2$gdppc), y = 0),
                 list(x = max(predictions2$gdppc), y = 0)),
    type = "line",
    color = "red",
    enableMouseTracking = FALSE,
    showInLegend = FALSE
  ) %>%
  hc_xAxis(title = list(text = "GDP per capita")) %>%
  hc_yAxis(title = list(text = "Residuals"))
```



	(1)
(Intercept)	110.123 (4.016)
gdppc	-0.015 (0.001)
gdp2	0.000 (0.000)
Num.Obs.	155
R2	0.656
R2 Adj.	0.652
AIC	1457.2
BIC	1469.3
Log.Lik.	-724.588
F	145.206
RMSE	25.94

The residuals exhibit less pattern than the linear model, but seem to overfit some of the nonlinearity in the sense that the residuals appear nonlinear and perhaps in the opposite direction of the observed data.

Multivariate Model: Combined Effects of Economics and Politics

Finally, let's examine how both economic development and political system jointly affect infant mortality rates. This multivariate model is somewhat more realistic insofar as we don't believe either of these variables acts alone in shaping infant mortality rates. Here, we'll use the logged GDP per capita measure as that functional form seemed to fit the data best (informally).

```
# Fit multivariate model
m5 <- lm(IMR ~ lngdp + polity, data=censor)
modelsummary(m4)
```

```
# Generate predictions for different political systems
gdp_range <- seq(min(censor$gdppc), max(censor$gdppc), length.out=100)
predictions_df <- data.frame(
  gdppc = rep(gdp_range, 2),
  lngdp = log(rep(gdp_range, 2)),
```

```

  polity = c(rep(-10, 100), rep(10, 100))
)

# Calculate predicted values
predictions_df$predicted <- predict(m5, newdata=predictions_df)

library(highcharter)
library(dplyr)

# Generate predictions for different political systems
gdp_range <- seq(min(censor$gdppc), max(censor$gdppc), length.out=100)
predictions_df <- data.frame(
  gdppc = rep(gdp_range, 2),
  lngdp = log(rep(gdp_range, 2)),
  polity = c(rep(-10, 100), rep(10, 100))
)

# Calculate predicted values
predictions_df$predicted <- predict(m5, newdata=predictions_df)

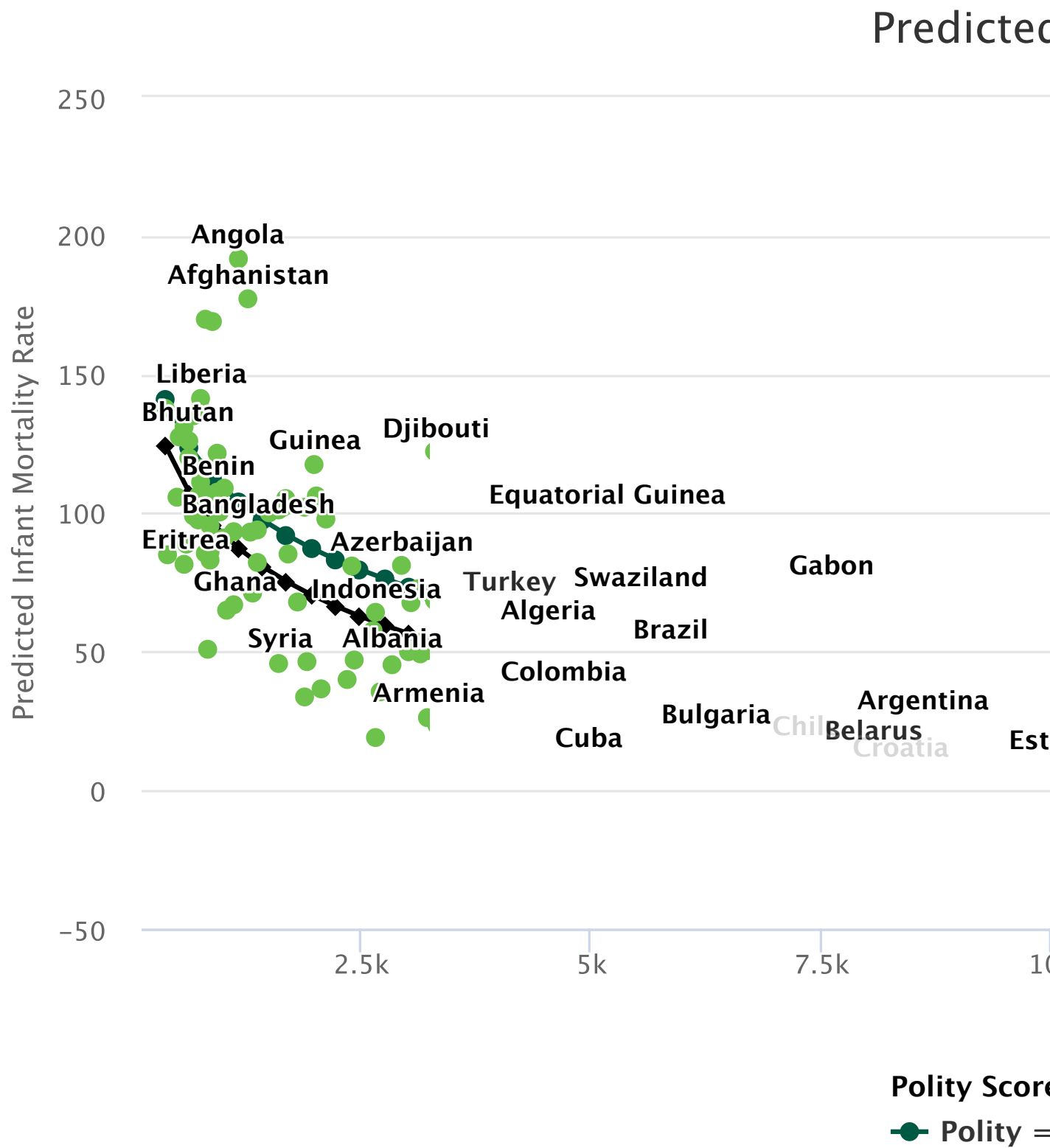
# Create highchart
highchart() %>%
  hc_add_series(data = predictions_df[predictions_df$polity == -10,],
    hcaes(x = gdppc, y = predicted),
    type = "line",
    color = "#005A43",
    name = "Polity = -10") %>%
  hc_add_series(data = predictions_df[predictions_df$polity == 10,],
    hcaes(x = gdppc, y = predicted),
    type = "line",
    color = "black",
    name = "Polity = 10") %>%
  hc_add_series(data = censor,
    hcaes(x = gdppc, y = IMR),
    type = "scatter",
    name = "Observed Values",
    marker = list(symbol = "circle"),
    color = "#6CC24A",
    dataLabels = list(enabled = TRUE, format = "{point.ctr}")) %>%
  # hc_add_series(data = censor,

```

```

#             hcaes(x = gdppc, y = IMR),
#             type = "scatter",
#             name = "Country Labels",
#             dataLabels = list(
#                 enabled = TRUE,
#                 format = "{point.ctr}",
#                 style = list(fontSize = "8px")
#             )) %>%
hc_xAxis(title = list(text = "GDP per capita")) %>%
hc_yAxis(title = list(text = "Predicted Infant Mortality Rate")) %>%
hc_title(text = "Predicted IMR by GDP and Political System") %>%
hc_legend(title = list(text = "Polity Score")) %>%
hc_tooltip(shared = FALSE)

```

This final plot shows how infant mortality rates are predicted to vary with GDP per capita for both highly autocratic (Polity = -10) and highly democratic (Polity = +10) countries. The actual observed values are shown as green points with country labels.