第21届国际关系研究方法研讨班

线性回归模型

陈冲
清华大学国际关系学系
cc458.github.io
2023年6月28日

Agenda

- Linear regression with one predictor
- Liner regression with multiple predictors
- Model Assumptions for OLS

Linear regression with one predictor

Regression analysis

- What is regression?
  - The term "regression" was used by Francis Galton in his 1886 paper "*Regression towards mediocrity in hereditary stature*"
  - **regression toward the mean** (biological phenomenon): the heights of descendants of tall ancestors tend to regress down towards a normal average.
- What is regression analysis?
  - a statistical method that allows you to examine the relationship between two or more variables of interest
  - to sort out which of those variables does indeed have an impact

Regression analysis: An example

- Why do a regression analysis?
  - To predict the value of a variable of interest
  - To make inference about the relationship between variables
- An example: **Does militarization affect economic development?**
- To answer this question, we will analyze the relationship between **militarization** and **economic development**

Regression analysis: An example

- How to measure **militarization** and **economic development**?
  - We use proxy measurement:
    - **militarization**: military expenditures as % of GDP
    - **economic development**: GDP per capita
  - We can obtain the data from World Bank's WDI
- Variables:
  - `gdppc`: GDP per capita (log)
  - `miliper`: military expenditure as % of GDP

Data and packages

```
library(tidyverse)
library(broom)
library(modelr)
library(knitr)
library(labelled) #add variable labels
library(cowplot) #plot_grid() function
library(car) # model assumption check
load(url("https://cc458.github.io/files/IRdata.RData"))
dim(IRdata)
```

```
## [1] 3687   13
```

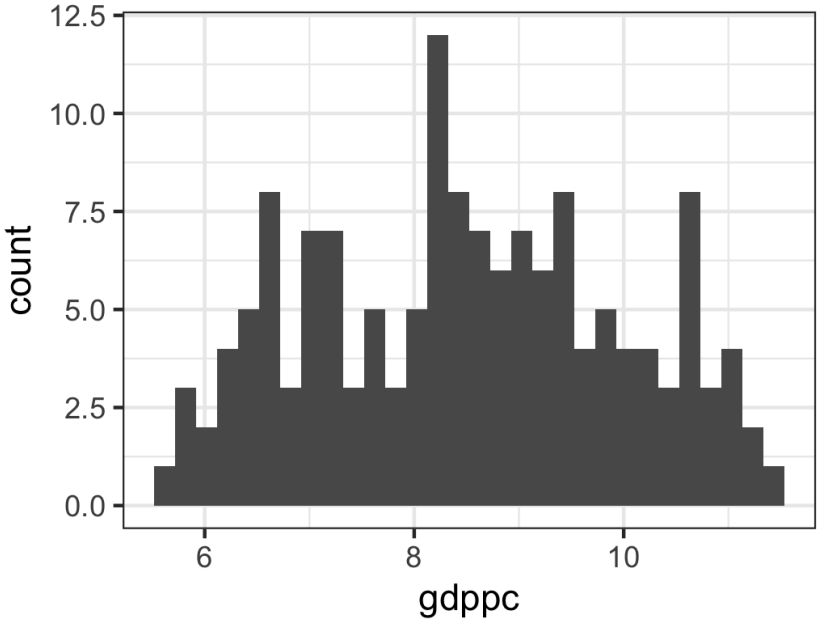
```
# we will use the 2015 data for now
df <- IRdata %>%
      filter(year == 2015)
```

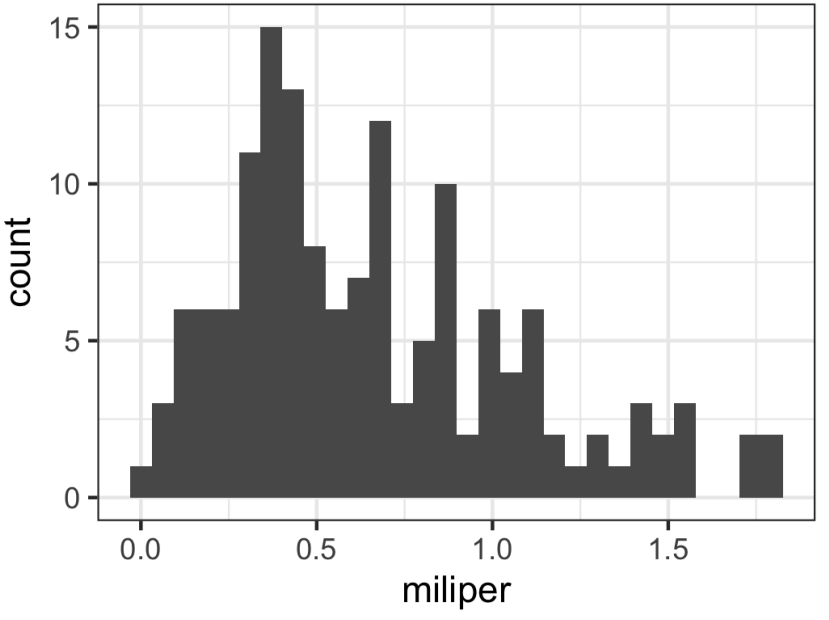
`glimpse` of the data

```
glimpse(df)
```

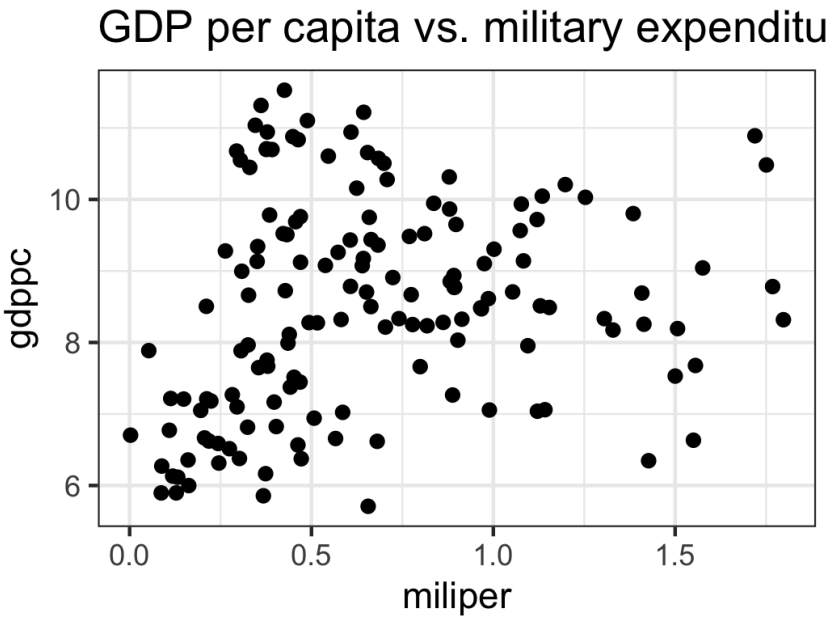
```
## Rows: 148
## Columns: 13
## $ ccode        <int> 2, 20, 40, 41, 42, 51, 52, 70, 90, 91, 92, 93, 94, 95, 1
## $ country_name <chr> "United States of America", "Canada", "Cuba", "Haiti", "
## $ year         <int> 2015, 2015, 2015, 2015, 2015, 2015, 2015, 2015, 2015, 20
## $ pop          <dbl> 19.58663, 17.39482, 16.25450, 16.18679, 16.16959, 14.870
## $ gdppc        <dbl> 10.941465, 10.676294, 8.936333, 6.703858, 8.774850, 8.50
## $ gdpgrowth    <dbl> 2.86158703, 0.94167586, 4.43833359, 1.21121834, 7.040936
## $ miliper      <dbl> 0.60870042, 0.29409085, 0.89154068, 0.00249989, 0.893912
## $ state_vio    <dbl> 14, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 28, 0, 0, 0,
## $ oneside_vio  <dbl> 0, 0, 0, 0, 0, 0, 0, 4, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0,
## $ vio          <dbl> 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1,
## $ polity2      <int> 10, 10, -7, 0, 8, 9, 10, 8, 8, 7, 8, 9, 10, 9, 7, 7, 5,
## $ polcomp      <int> 10, 10, 1, -77, 9, 9, 10, 9, 8, 9, 9, 9, 10, 10, 7, 8, 8
## $ dem          <dbl> 3, 3, 1, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 2, 2, 3,
```

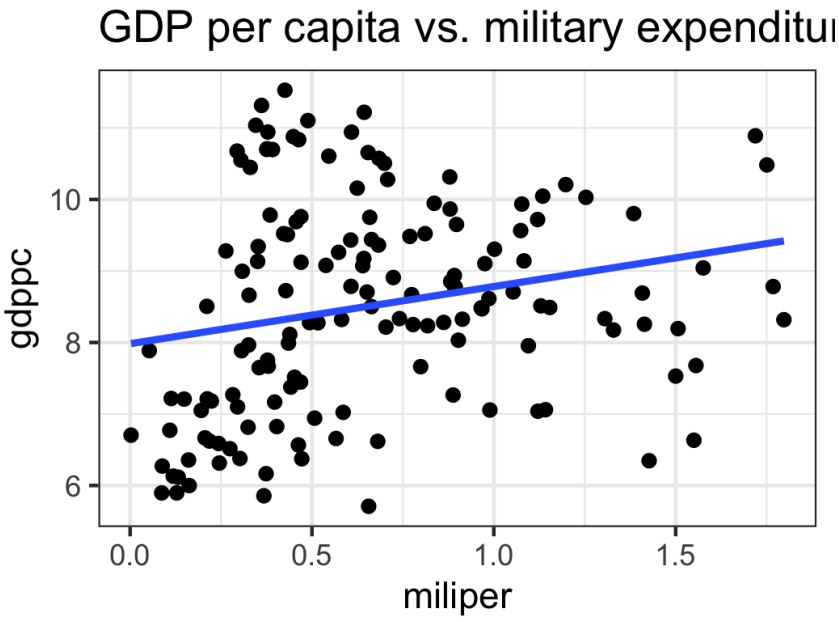
```
p1 <- ggplot(data = df ,mapping = aes(x = gdppc)) +
  geom_histogram()
p2 <- ggplot(data = df, mapping = aes(x = miliper)) +
  geom_histogram()
plot_grid(p1, p2, ncol = 2)
```





```
p1 <- ggplot(data = df, aes(x = miliper, y = gdppc)) + geom_point() +
  labs(title = "GDP per capita vs. military expenditures")
p2 <- ggplot(data = df, mapping = aes(x = miliper, y = gdppc)) + geom_point()
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "GDP per capita vs. military expenditures")
plot_grid(p1, p2, ncol = 2)
```





Terminology

- **GDP per capita**(`gdppc`) is the response variable (Y)
  - variable whose variation we want to understand and/or variable we wish to predict
  - also known as *dependent*, *outcome*, *target*, *output* variable (因变量、结果变量、目标变量，输出变量等)

- **Military expenditures**(`miliper`) is the predictor variable (X)
  - variable used to account for variation in the response
  - also known as *independent*, *explanatory*, *input* variable(自变量、解释变量、输入变量)

Model

$$\text{GDP per capita} = f(\text{military expenditures}) + \epsilon$$

We want to estimate f. How do we do it? A general form of the model:

$$Y = f(\mathbf{X}) + \epsilon$$

- Y: quantitative response variable
- $\mathbf{X} = (X_1, X_2, \ldots, X_p)$: predictor variables
- f: fixed but unknown function
  - systematic information $\mathbf{X}$ provides about Y
- ϵ: random error term with mean 0 that is independent of $\mathbf{X}$

How to estimate f?

In general, we will use the following steps to estimate f

- Choose the functional form of f, i.e. choose the appropriate model given the data
  - Ex: f is a linear model

$$f(\mathbf{X}) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \epsilon$$

- Use the data to fit the model, i.e. estimate the model parameters
  - Ex: Use a method to estimate the model parameters $\beta_0, \beta_1, \ldots, \beta_p$

Why estimate f?

Suppose we have the model

$$\textbf{GDP per capita} = \beta_0 + \beta_1 \times \textbf{military expenditures} + \epsilon$$

There are two types of questions we may wish to answer using our model:

- Prediction: What is the expected Y given particular values of $X_1, X_2, \ldots, X_p$? - Ex: What is the expected GDP per capita for a country whoes military expenditure accounts for 5% of its total GDP?

- Inference: What is the relationship between $\mathbf{X}$ and Y. How does Y change as a function of $\mathbf{X}$? - Ex: How much can we expect GDP per capita to change for each additional percentage in the military expenditure?

Linear regression (线性回归)

- There is some true relationship between X and Y that exists in the population

$$Y = f(X) + \epsilon$$

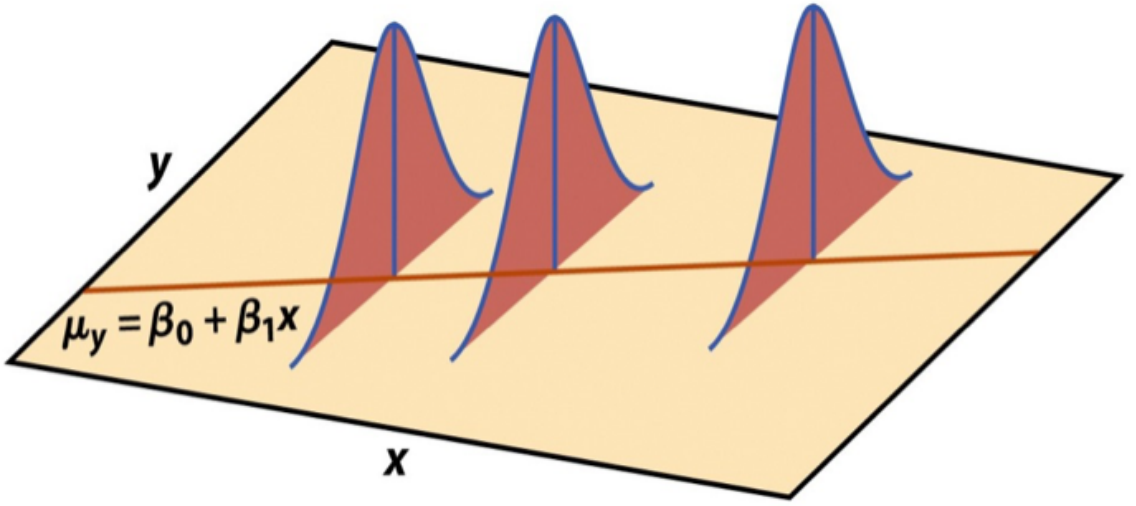
- If f is approximated by a linear function, then we can write the relationship as

$$Y = \beta_0 + \beta_1 X + \epsilon$$

- We'll use statistical inference to determine if the relationship we observe in the data is statistically significant or if it's due to random chance.

Regression model: $Y = \beta_0 + \beta_1 X + \epsilon$

$$Y|X \sim N(\beta_0 + \beta_1 X, \sigma^2)$$



- For any fixed x, y follows a normal distribution with a standard deviation of σ
- σ: the standard deviation of Y as a function of X; ***Assumption***: σ is equal for all values of X (equal variance of y)

Linear regression model

$$Y|X \sim N(\beta_0 + \beta_1 X, \sigma^2)$$

- For a single observation (x_i, y_i)

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i \quad \epsilon_i \sim N(0, \sigma^2)$$

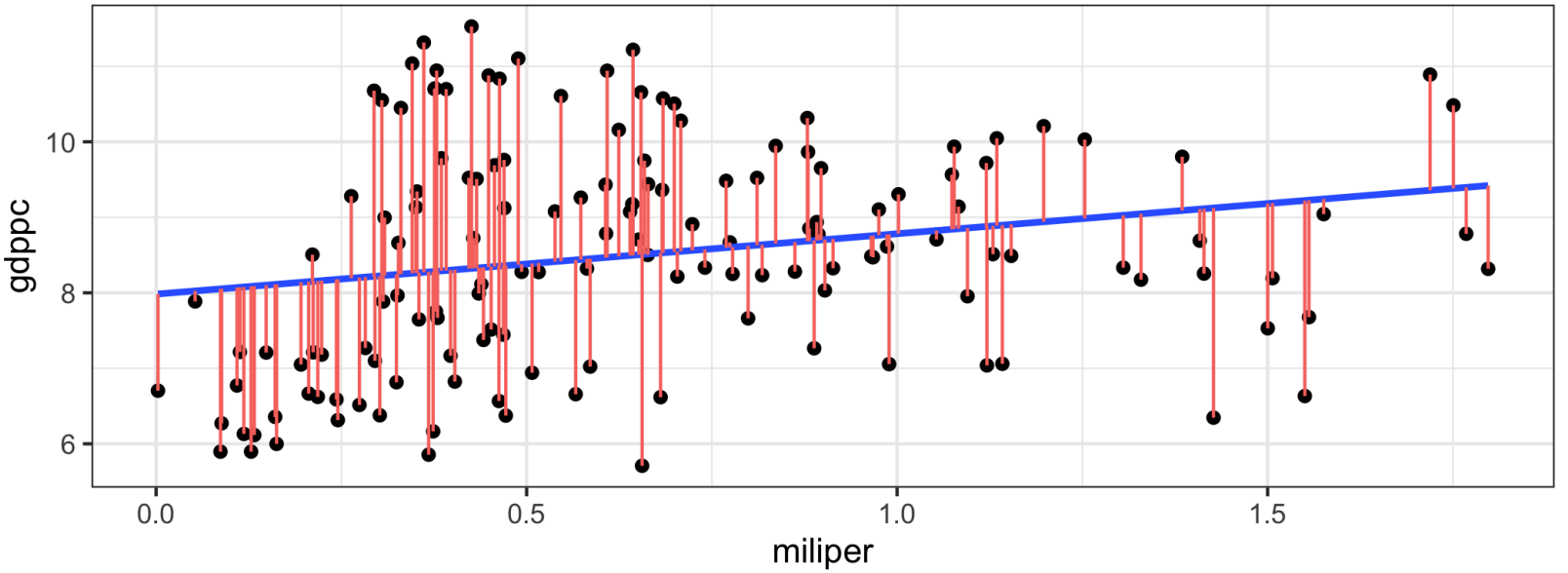
- We want to use the n observations $(x_1, y_1), \ldots, (x_n, y_n)$ to estimate β_0 and β_1.

- We'll use least-squares regression(最小二乘法) estimates.

  - *The Least Squares Regression line* is the line that makes the vertical distance from the data points to the regression line as small as possible. It's called a "least squares" because the best line of fit is one that minimizes the variance (the sum of squares of the errors)

Residuals

The residual is the difference between the observed and predicted values.



Residual sum of squares

- The residual for the i^{th} observation is

$$e_i = y_i - \hat{y}_i = y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i)$$

- The *residual sum of squares* is

$$RSS = e_1^2 + e_2^2 + \cdots + e_n^2$$

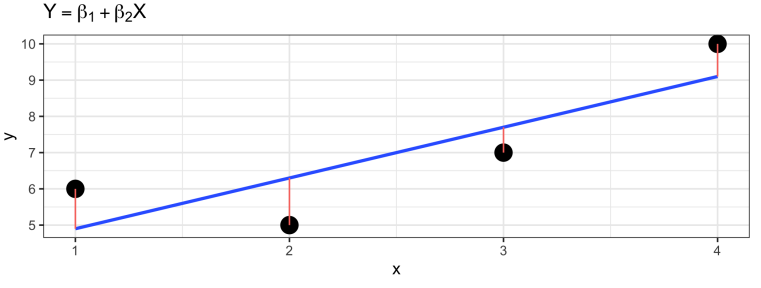
- The least-squares regression approach chooses coefficients $\hat{\beta}_0$ and $\hat{\beta}_1$ to minimize RSS.

- Note the difference between: $Y = \beta_0 + \beta_1 X + \epsilon$ and $\hat{Y} = \hat{\beta}_0 + \hat{\beta_1} X$

A toy example for linear least squares by hand: I

- Four (x, y) data point: $(1, 6)$, $(2, 5)$, $(3, 7)$, and $(4, 10)$
- We hope to find a line $Y = \beta_1 + \beta_2 X$ that best fits these four points.

points	x	y
point 1	1	6
point 2	2	5
point 3	3	7
point 4	4	10



A toy example for linear least squares by hand: II

- That is, to find β_1 and β_2 to approximately solve the overdetermined linear system:

$$\begin{aligned}\beta_1 + 1\beta_2 + \epsilon_1 &= 6\\ \beta_1 + 2\beta_2 + \epsilon_2 &= 5\\ \beta_1 + 3\beta_2 + \epsilon_3 &= 7\\ \beta_1 + 4\beta_2 + \epsilon_4 &= 10\end{aligned}$$

- The residual, at each point, between the `best curve fit` and the data is the difference between the right- and left-hand sides of the equations. The least squares approach to solving this problem is to try to make the **sum of the squares of these residuals** as small as possible;

$$RSS = e_1^2 + e_2^2 + \cdots + e_n^2$$

that is, to find the minimum of the function:

$$\begin{aligned}S(\beta_1, \beta_2) &= [6 - (\beta_1 + 1\beta_2)]^2 + [5 - (\beta_1 + 2\beta_2)]^2 + [7 - (\beta_1 + 3\beta_2)]^2 + [10 - \beta_1 + 4\beta_2\\ &= 4\beta_1^2 + 30\beta_2^2 + 20\beta_1\beta_2 - 56\beta_1 - 154\beta_2 + 210\end{aligned}$$

A toy example for linear least squares by hand: III

$$S(\beta_1, \beta_2) = 4\beta_1^2 + 30\beta_2^2 + 20\beta_1\beta_2 - 56\beta_1 - 154\beta_2 + 210$$

The minimum is determined by calculating the **partial derivatives**(偏导数) of $S(\beta_1, \beta_2)$ with respect to β_1 and β_2 and setting them to zero ("with the others held constant"):

$$\frac{\partial S}{\partial \beta_1} = 4 * 2\beta_1 + 0 + 20\beta_2 * 1 - 56 - 0 + 0 = 8\beta_1 + 20\beta_2 - 56 = 0$$
$$\frac{\partial S}{\partial \beta_2} = 0 + 30 * 2\beta_2 + 20\beta_1 - 0 - 154 + 0 = 60\beta_2 + 20\beta_1 - 154 = 0$$

solve a system of two equations:

$$8\beta_1 + 20\beta_2 - 56 = 0$$
$$20\beta_1 + 60\beta_2 - 154 = 0$$

$\beta_1 = 3.5$; $\beta_2 = 1.4$, and $y = 3.5 + 1.4 \times x$

Minimum residuals:
$S(\beta_1, \beta_2) = S(\beta_1 = 3.5, \beta_2 = 1, 5) = 1.1^2 + (-1.3)^2 + (-0.7)^2 + 0.9^2 = 4.2$

Estimating coefficients of least squares models

- Slope:

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} = r\frac{s_y}{s_x}$$

such that r is the correlation between x and y.

- Intercept: $\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$

How to find the linear regression line by hand?

Table: Data

ccode	gdppc	miliper	Xbar	Ybar	X-Xbar	Y-Ybar	(X-Xbar) (Y-Ybar)	Xb
2	10.941465	0.6087004	0.6646869	8.514311	-0.0559865	2.4271536	-0.1358877	0.00
20	10.676293	0.2940909	0.6646869	8.514311	-0.3705960	2.1619822	-0.8012220	0.13
40	8.936333	0.8915407	0.6646869	8.514311	0.2268538	0.4220212	0.0957371	0.05
41	6.703858	0.0024999	0.6646869	8.514311	-0.6621870	-1.8104530	1.1988584	0.43
42	8.774850	0.8939124	0.6646869	8.514311	0.2292256	0.2605384	0.0597221	0.05
51	8.505290	0.2109311	0.6646869	8.514311	-0.4537558	-0.0090212	0.0040934	0.20

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}; \ \hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

```
sum(ols$`(X-Xbar)(Y-Ybar)`);sum(ols$`(X-Xbar)^2`)
```

```
## [1] 20.06906
```

```
## [1] 25.09528
```

```
(beta_1 <- sum(ols$`(X-Xbar)(Y-Ybar)`)/sum(ols$`(X-Xbar)^2`))
```

```
## [1] 0.7997146
```

```
(beta_0 <- mean(ols$gdppc) - beta_1*mean(ols$miliper))
```

```
## [1] 7.982752
```

Least squares model

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} = r\frac{s_y}{s_x}$$

```
y <- df$gdppc; x <- df$miliper
r <- cor(x, y)
(beta_1 <- r*(sd(y)/sd(x)))
```

```
## [1] 0.7997146
```

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1\bar{x}$$

```
(beta_0 <- mean(y) - beta_1 * mean(x))
```

```
## [1] 7.982752
```

Fit a least squares model with R

In **R**, we use the `lm()` function to fit a least-squares model

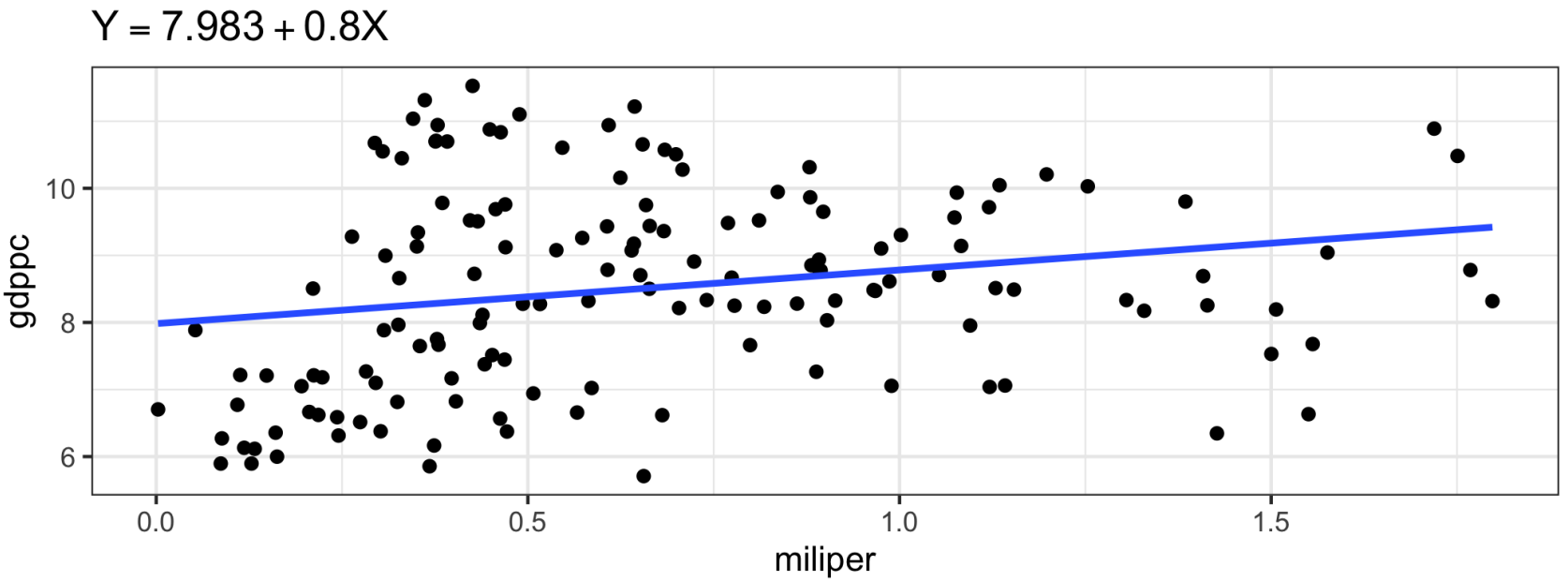
```
lm(formula, data, subset, weights, na.action,
   method = "qr", model = TRUE, x = FALSE, y = FALSE, qr = TRUE,
   singular.ok = TRUE, contrasts = NULL, offset, ...)
```

`formula: y ~ x`

```
model <- lm(gdppc ~ miliper, data = df)
tidy(model) %>% kable(format = "markdown", digits = 3)
```

term	estimate	std.error	statistic	p.value
(Intercept)	7.983	0.225	35.456	0.000
miliper	0.800	0.288	2.777	0.006

$$\widehat{\text{GDP per capita}} = 7.983 + 0.800 \times \text{military expenditures}$$


Y = 7.983 + 0.8X
gdppc
10
8
6
0.0
0.5
1.0
1.5
miliper

Interpreting slope & intercept

- Slope: Increase in the mean response for every one unit increase in the predictor variable
- Intercept: Mean response when the explanatory variable equals 0

Coefficient of determination, R-squared

The coefficient of determination, denoted R^2, is a statistical measure that represents the proportion of the variance for a dependent variable that's explained by an independent variable or variables in a regression model.

$$R^2 = 1 - \frac{\text{sum of squares of residuals (RSS)}}{\text{total sum of squares(TSS)}} = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}$$

where $\hat{y}_i$ is the predicted value of y_i (or fitted value of y_i)

```
(r_squared = 1- sum((df$gdppc - model$fitted.values)^2)/sum((df$gdppc - mea
```

```
## [1] 0.05018082
```

- As the number of independent variables increases, the R^2 never deceases
  - We use the adjusted R^2 instead, which penalizes the

Nonsensical intercept

- Sometimes it doesn't make sense to interpret the intercept
  - When predictor variable doesn't take values close to 0
  - When the intercept is negative even though the response variable should always be positive
- The intercept helps the line fit the data as closely as possible
- It is fine to have a nonsensical intercept if it helps the model give better overall predictions

Export regression outcome: table

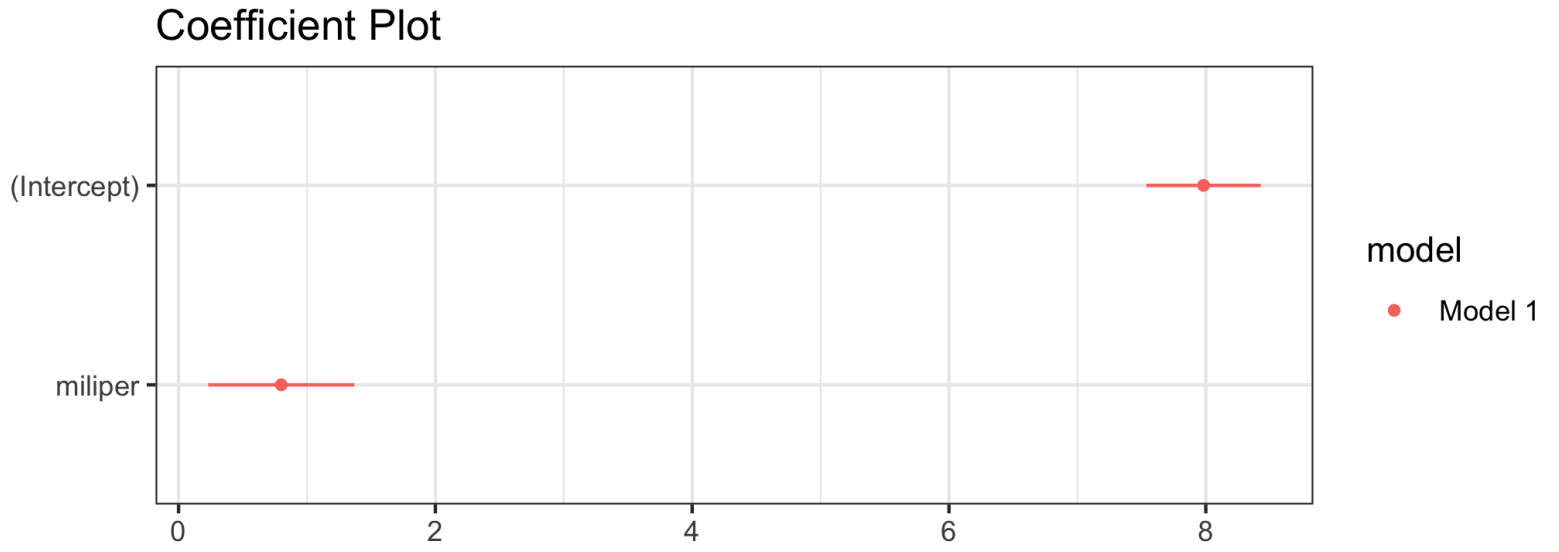
```
library(stargazer)
stargazer(model, type = "html", title="Regression Results", single.row=TRUE,
```

Regression Results

	Dependent variable:
	gdppc
miliper	0.800*** (0.235, 1.364)
Constant	7.983*** (7.541, 8.424)
Observations	148
R^2	0.050
Adjusted R^2	0.044
Residual Std. Error	1.442 (df = 146)
F Statistic	7.713*** (df = 1; 146)
Note:	*p<0.1; **p<0.05; ***p<0.01

Export regression outcome: graph

```
library(dotwhisker)
dwplot(list(model), conf.level = .95, show_intercept = TRUE,
       model_name = "model 1",) + theme_bw() + ggtitle("Coefficient Plot")
```



Multiple Linear Regression

Questions

- What is the relationship between the characteristics of a country and economic development?
- Given its characteristics, what is the expected level of economic development?

Variables

Predictors

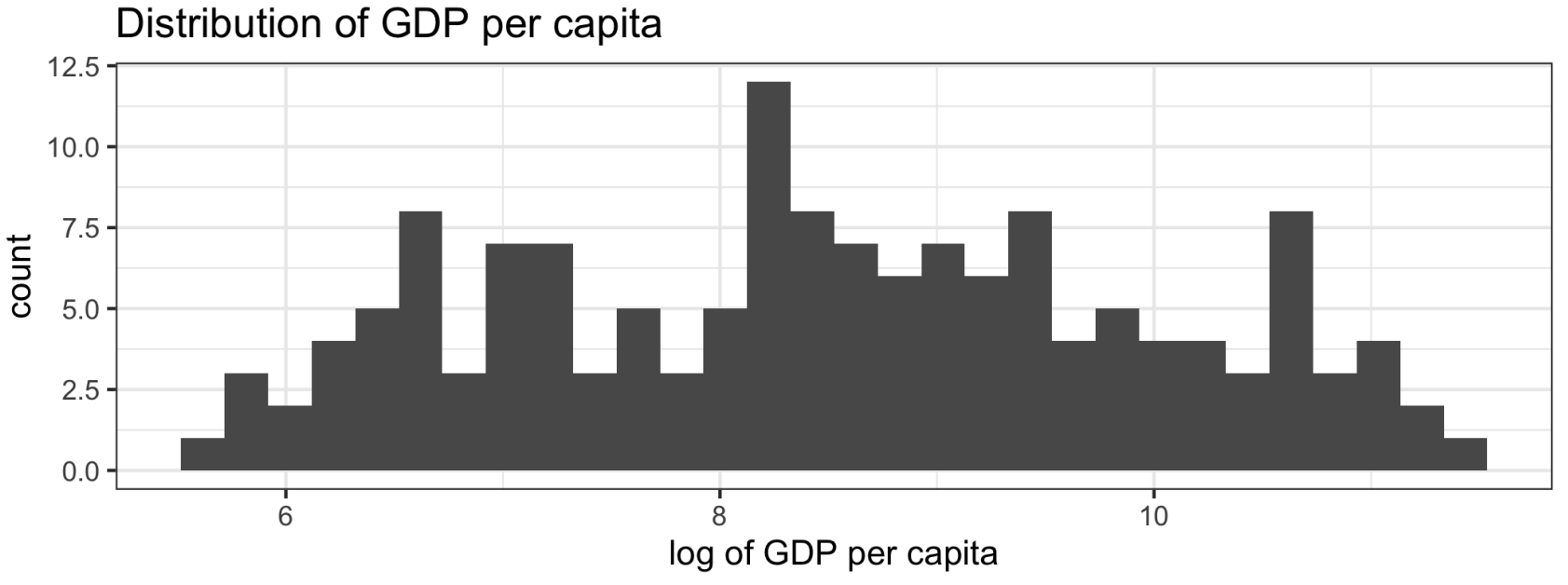
- `miliper`: military expenditures as % of GDP
- `pop`: Number of total population(log)
- `polity2`: regime type scores [-10, 10]
- `vio`: whether there was political violence

Response

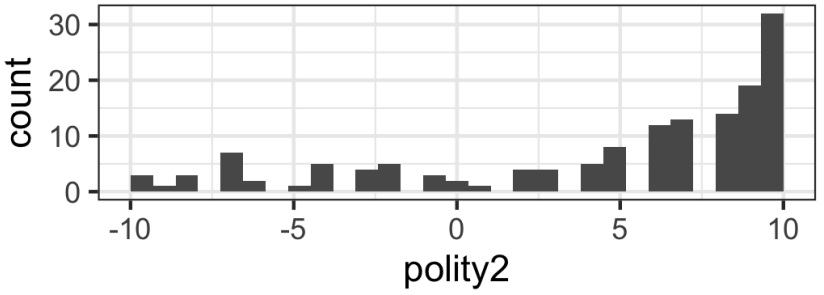
- `gdppc`: log of GDP per capita (in U.S. dollars)

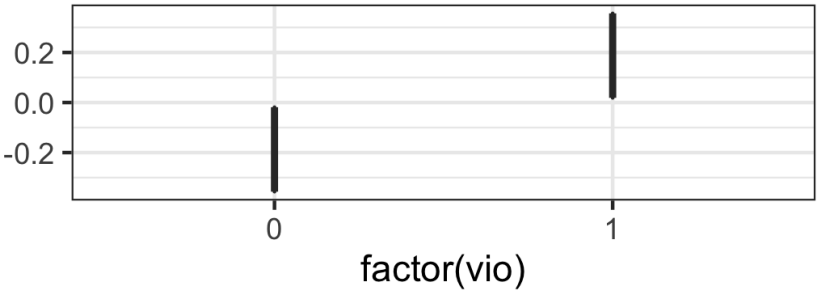
EDA: Response variable

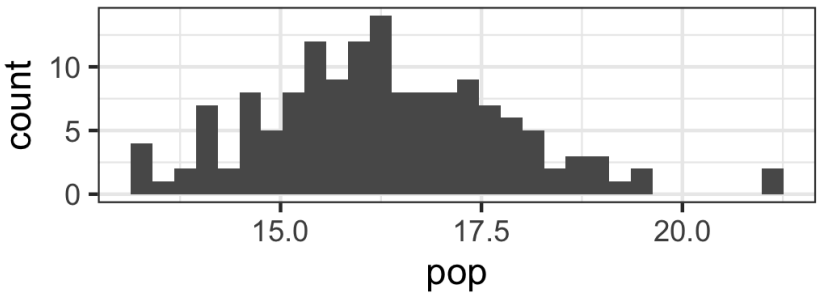
```
ggplot(data = df, aes(x = gdppc)) + geom_histogram() +
  labs(title = "Distribution of GDP per capita", x = "log of GDP per capita")
```

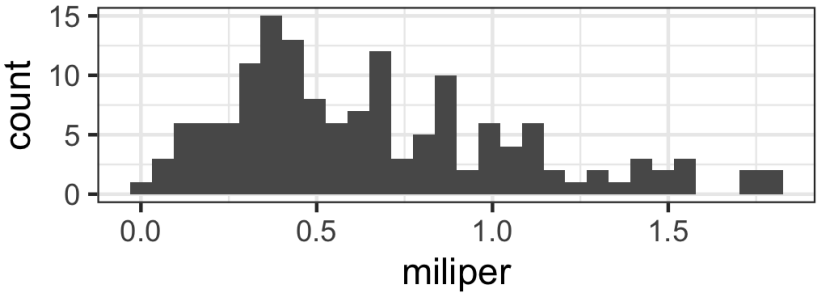


EDA: Predictor variables



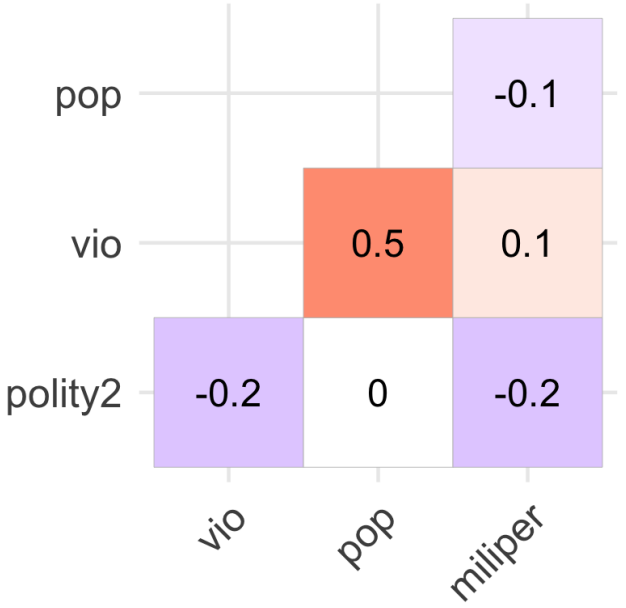




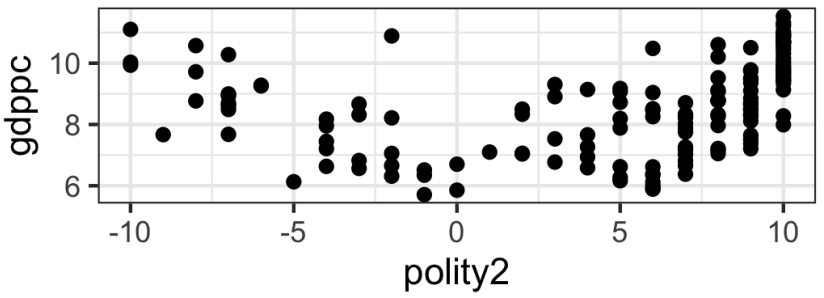


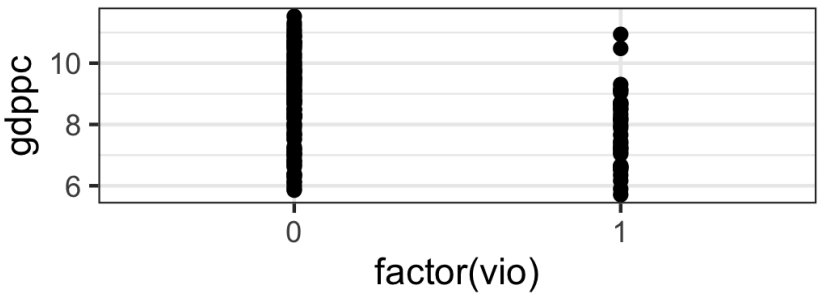
Correlation matrix of independent variables

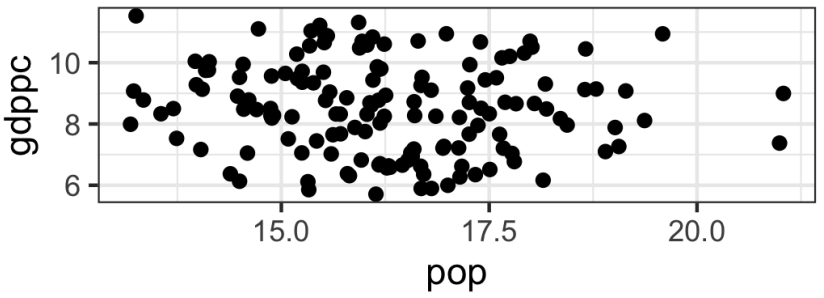
```
library(ggcorrplot)
df %>% dplyr::select(polity2, vio, pop, miliper) %>%
  cor(use = "pairwise") %>%
  round(1) %>%
  ggcorrplot(., type = "lower", lab = T, show.legend = F)
```

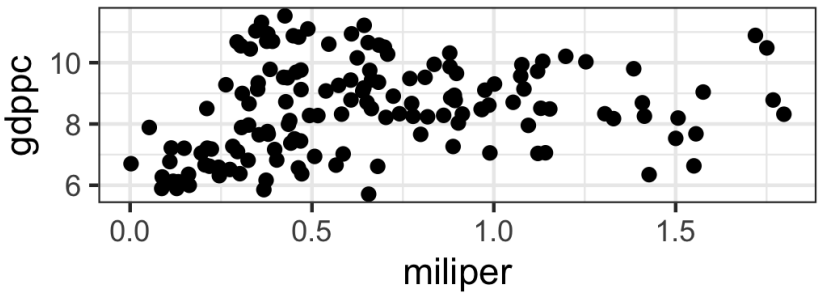

pop
vio
polity2
-0.1
0.5
0.1
-0.2
0
-0.2
vio
pop
miliper

EDA: Response vs. Predictors









What is a disadvantage to fitting a separate model for each predictor variable?

Multiple regression model

We will calculate a multiple linear regression model with the following form:

$$\begin{aligned}\text{GDP per capita} = \beta_0 &+ \beta_1 \text{polity2} + \beta_2 \text{violence} + \beta_3 \text{population} \\ &+ \beta_4 \text{military expenditures} + \epsilon\end{aligned}$$

Similar to simple linear regression, this model assumes that at each combination of the predictor variables, the values **GDP per capita** follow a Normal distribution

Regression model

- Recall: The simple linear regression model assumes

$$y|x \sim N(\beta_0 + \beta_1 x, \sigma^2)$$

- Similarly: The multiple linear regression model assumes

$$y|x_1, x_2, \ldots, x_p \sim N(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p, \sigma^2)$$

For a given observation $(x_{i1}, x_{i2} \ldots, x_{iP}, y_i)$

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} + \epsilon_i \quad \epsilon_i \sim N(0, \sigma^2)$$

Regression model

- At any combination of $x's$, the true mean value of y is

$$\mu_y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p$$

- We will use multiple linear regression to estimate the mean y for any combination of $x's$

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \cdots + \hat{\beta}_p x_p$$

Regression output I

```
model2 <- lm(gdppc ~ polity2 + vio + pop + miliper, data = df)

tidy(model2, conf.int = TRUE) %>%
  kable(format = "markdown", digits = 3)
```

term	estimate	std.error	statistic	p.value	conf.low	conf.high
(Intercept)	5.987	1.393	4.297	0.000	3.233	8.741
polity2	0.063	0.019	3.249	0.001	0.025	0.101
vio	-1.112	0.302	-3.685	0.000	-1.708	-0.516
pop	0.106	0.085	1.255	0.211	-0.061	0.273
miliper	1.179	0.277	4.250	0.000	0.630	1.727

Regression output II: table

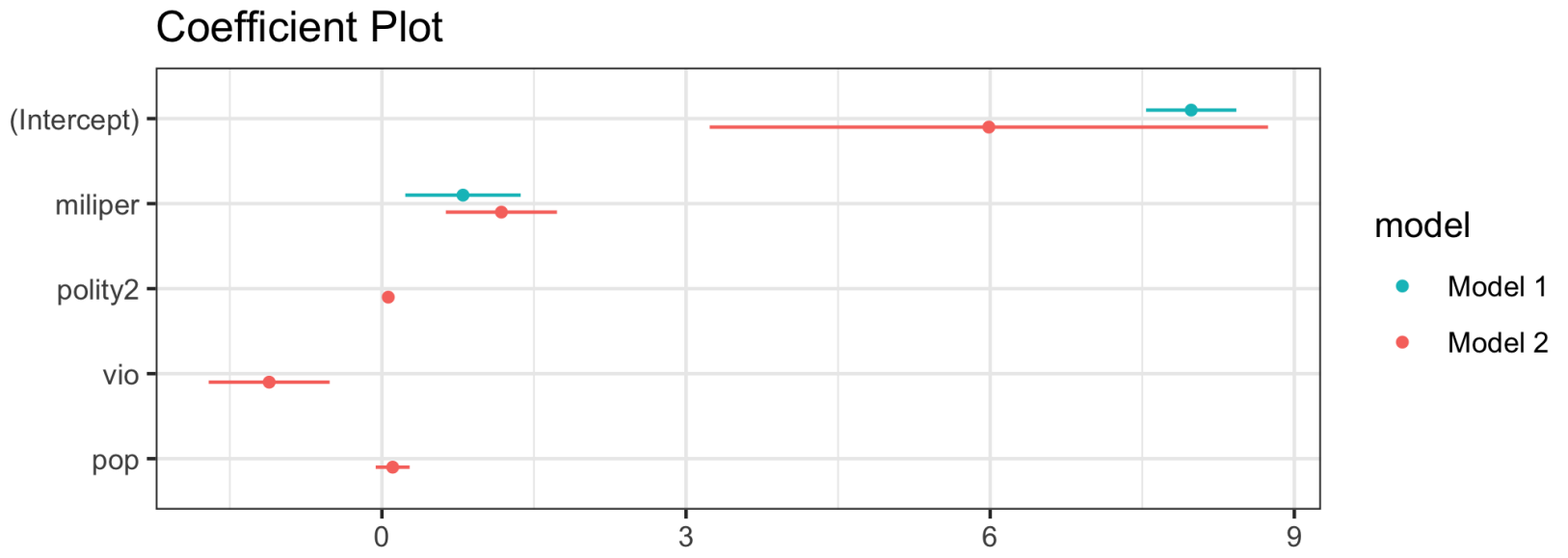
```
stargazer(model, model2, type = "html", title="Regression Results",
          single.row=TRUE, ci=TRUE, ci.level=0.95)
```

Regression Results

	Dependent variable:	
	gdppc	
	(1)	(2)
polity2		0.063*** (0.025, 0.101)
vio		-1.112*** (-1.703, -0.521)
pop		0.106 (-0.060, 0.272)
miliper	0.800*** (0.235, 1.364)	1.179*** (0.635, 1.722)
Constant	7.983*** (7.541, 8.424)	5.987*** (3.256, 8.717)
Observations	148	148
R^2	0.050	0.212
Adjusted R^2	0.044	0.190
Residual Std. Error	1.442 (df = 146)	1.327 (df = 143)
F Statistic	7.713*** (df = 1; 146)	9.635*** (df = 4; 143)

Regression output II: graph

```
dwplot(list(model, model2), conf.level = .95, show_intercept = TRUE) +
  theme_bw() + ggtitle("Coefficient Plot")
```



Interpreting $\hat{\beta}_j$

- An estimated coefficient $\hat{\beta}_j$ is the expected change in y to change when x_j increases by one unit **holding the values of all other predictor variables constant**.

- *Example:*
  - The estimated coefficient for **polity2** is 0.063. This means for each additional point of polity score, we expect the log of of GDP per capita to increase by 0.063 (that is, `exp(0.063)` =1.065027), on average, holding all other predictor variables constant.

Hypothesis tests for $\hat{\beta}_j$

- We want to test whether a particular coefficient has a value of 0 in the population, given all other variables in the model:

$$H_0 : \beta_j = 0$$
$$H_a : \beta_j \neq 0$$

- The test statistic reported in R is the following:

$$\text{test statistic} = t = \frac{\hat{\beta}_j - 0}{SE(\hat{\beta}_j)}$$

- Calculate the p-value using the t distribution with $n - p - 1$ degrees of freedom, where p is the number of terms in the model (not including the intercept).

Inference in multiple linear models

- β_j has a t Student distribution
- We can make inference for multiple linear constrains: $H_0 : \beta_1 = \beta_2 = \ldots \beta_j = 0$ through F test

$F = \frac{(RSS_r - RSS_c)}{RSS_c/(n-k-1)}$ where c is the complete model, and r is the restricted model

```
anova(model2, model)
```

```
## Analysis of Variance Table
##
## Model 1: gdppc ~ polity2 + vio + pop + miliper
## Model 2: gdppc ~ miliper
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    143 251.93
## 2    146 303.78 -3   -51.852 9.8107 6.313e-06
```

Confidence interval for β_j

The confidence interval for β_j

$$\hat{\beta}_j \pm t^* SE(\hat{\beta}_j)$$

where t^* follows a t distribution with with $(n - p - 1)$ degrees of freedom

- **General Interpretation**: We are C confident that the interval lower bound to upper bound contains the population coefficient of x_j. Therefore, for every one unit increase in x_j, we expect y to change by LB to UB units, holding all else constant.

Confidence interval for `political violence`

Interpret the 95% confidence interval for the coefficient of `vio`.

term	estimate	std.error	statistic	p.value	conf.low	conf.hig
(Intercept)	5.987	1.393	4.297	0.000	3.233	8.74
polity2	0.063	0.019	3.249	0.001	0.025	0.10
vio	-1.112	0.302	-3.685	0.000	-1.708	-0.51
pop	0.106	0.085	1.255	0.211	-0.061	0.27
miliper	1.179	0.277	4.250	0.000	0.630	1.72

Caution: Large sample sizes

If the sample size is large enough, the test will likely result in rejecting $H_0 : \beta_j = 0$ even x_j has a very small effect on y

- Consider the practical significance of the result not just the statistical significance
- Use the confidence interval to draw conclusions instead of p-values

Caution: Small sample sizes

If the sample size is small, there may not be enough evidence to reject $H_0 : \beta_j = 0$

- When you fail to reject the null hypothesis, **DON'T** immediately conclude that the variable has no association with the response.
- There may be a linear association that is just not strong enough to detect given your data, or there may be a non-linear association.

Caution: "control for another variable in multiple regression?"

- It is wrong to say **"we control for another variable"**. You should say **we control the effect of other variable(s)**, which means we remove the effect of other variables from the relation between the two or more variables. This implies that we keep the effect of other variables brought explicitly in the model constant.

- What does it mean to control for the variables in the model? It means that when you look at the effect of one variable in the model, you are holding constant all of the other predictors in the model.

Prediction

- We calculate predictions the same as with simple linear regression

```
tidy(model2) %>% select(term,estimate) %>% t()
```

```
##          [,1]          [,2]          [,3]          [,4]          [,5]
## term     "(Intercept)" "polity2"     "vio"         "pop"         "miliper"
## estimate " 5.98686841" " 0.06277831" "-1.11200537" " 0.10622131" " 1.17858598"
```

- **Example:** What is the predicted log of GDP per capita for a country with `polity2 = 10`, `violence = TRUE`, `pop = 10`, `miliper = 3`%?

```
5.987 + 0.063 * 10 -1.112 * 1 + 0.106 * 10  + 1.179 * 3
```

```
## [1] 10.102
```

- The predicted GDP per capita is:

```
exp(5.987 + 0.063 * 10 -1.112 * 1 + 0.106 * 10  + 1.179 * 3 )
```

```
## [1] 24391.74
```

Intervals for predictions

- Just like with simple linear regression, we can use the `predict` function in R to calculate the appropriate intervals for our predicted values

```
x0 <- data.frame(polity2 = 10, vio = 1, pop = 10, miliper = 3)
predict(model2, x0, interval = "prediction")
```

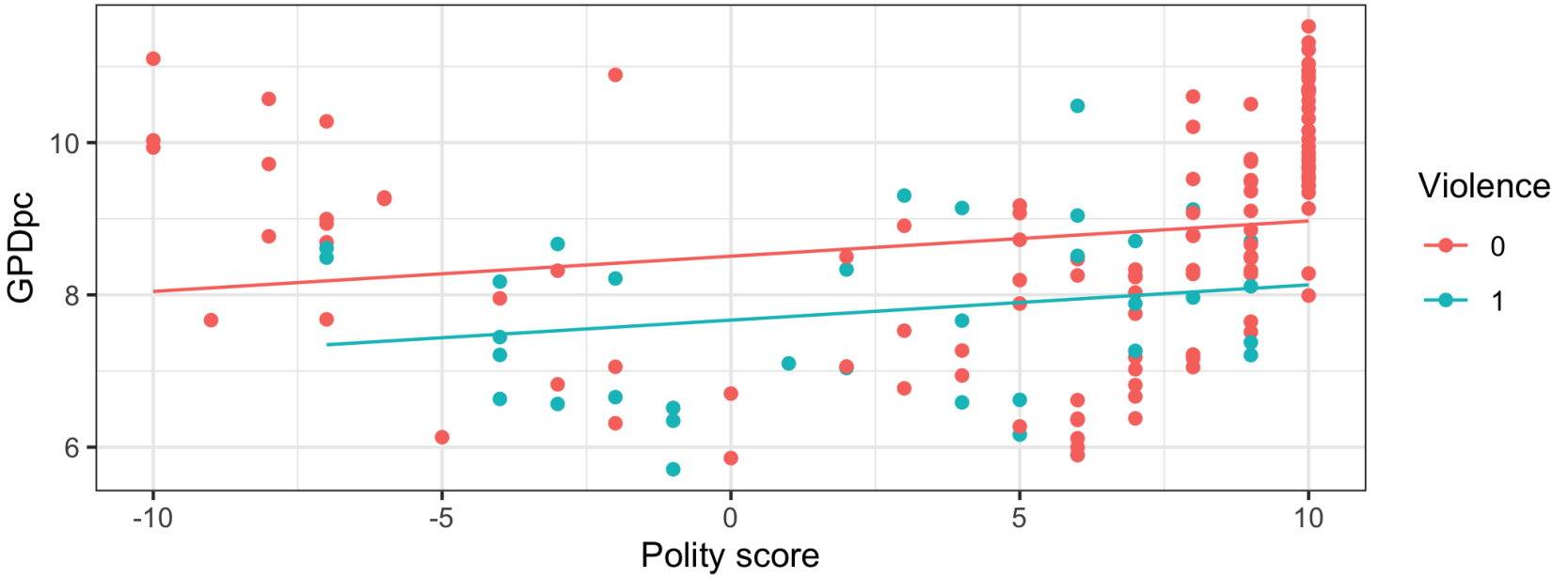
```
##        fit      lwr      upr
## 1 10.10062 6.952866 13.24837
```

```
exp(predict(model2, x0, interval = "prediction"))
```

```
##        fit      lwr    upr
## 1 24358.04 1046.144 567144
```

Predicted values

```
library(prediction)
model2 <- lm(gdppc ~ polity2 + vio, data = df)
pred_model_2 <- as_tibble(prediction(model2))
ggplot(data = pred_model_2) + # the new predicted values
geom_point(mapping = aes(x = polity2, y = gdppc,
          color = factor(vio))) +
# the regression lines are drawn (differentiated by color):
geom_line(mapping = aes(x = polity2, y = fitted, color = factor(vio),
                        group = factor(vio))) +
  labs(x = "Polity score", y = "GPDpc", color = "Violence")
```


GPDpc
10
8
6
-10
-5
0
5
10
Polity score
Violence
0
1

Cautions

- Do not extrapolate! Because there are multiple explanatory variables, you can extrapolation in many ways
- The multiple regression model only shows association, not causality
  - To show causality, you must have a carefully designed experiment or carefully account for confounding variables in an observational study

Checking Model Assumptions for OLS

Statistical Models

> "Essentially all models are wrong, but some are useful."
>
> George Box, "Science and Statistics," *Journal of the American Statistical Association*, 1976.

- Models are simplified and idealized representations of systems or objects:
  - Models will never be "the truth" if truth means entirely representative of reality
- Because they are simplified, models are often helpful in understanding a certain component of a system

Assumptions for Regression

1. Linearity: The plot of the mean value for y against x falls on a straight line

2. Constant Variance: The regression variance is the same for all values of x (homoscedasticity, v.s., heteroscedasticity)

3. Normality: For a given x, the distribution of y around its mean is Normal

4. Independence: All observations are independent

Checking Assumptions

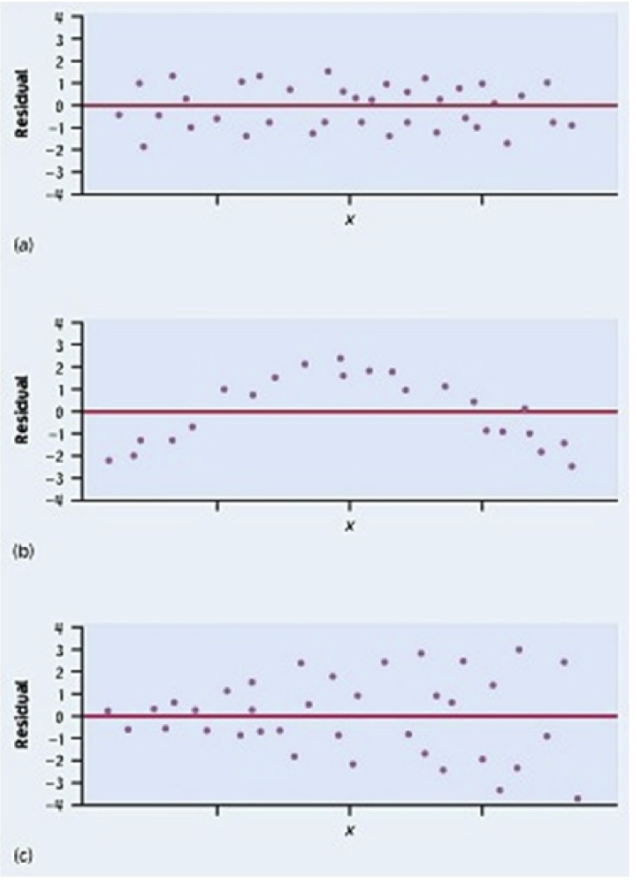
We can use plots of the residuals to check the assumptions for regression.

1. Scatterplot of Y vs. X (linearity).
    - Check this before fitting the regression model.
2. Plot of residuals vs. predictor variable (constant variance, linearity)
3. Histogram and Normal QQ-Plot of residuals (Normality)

Residuals vs. Predictor

- When all the assumptions are true, the values of the residuals reflect random (chance) error
- We can look at a plot of the **residuals** vs. **the predictor variable**
- There should be no distinguishable pattern in the residuals plot, i.e. the residuals should be randomly scattered
- A non-random pattern suggests assumptions might be violated

Plots of Residuals



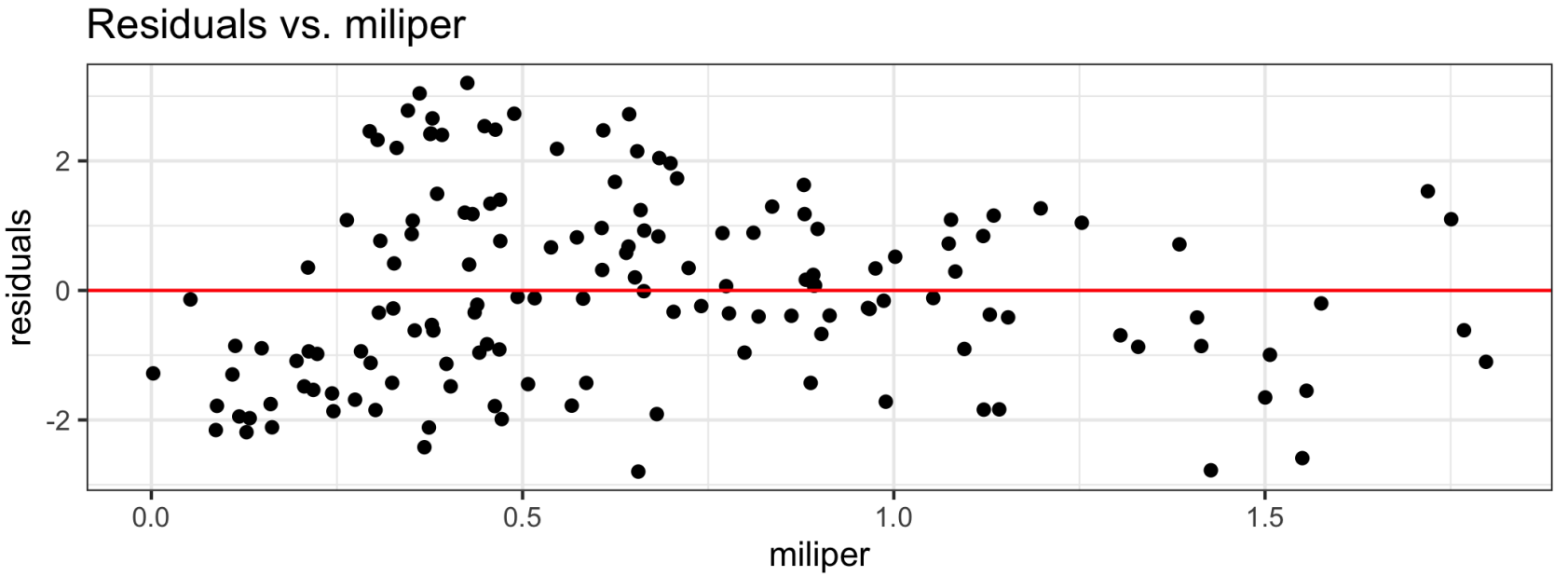
Ideal Residual Plot

Nonlinearity

Nonconstant Variance

Checking assumptions for constant variance and linearity with R

```
df <- df %>% mutate(residuals=resid(model))
ggplot(data = df) +
  geom_point(aes(x = miliper, y = residuals)) +
  geom_hline(yintercept=0,color="red")+
  labs(title="Residuals vs. miliper")
```



Statistical diagnosis

```
library(lmtest)
bptest(model, studentize = T)
```

```
##
##     studentized Breusch-Pagan test
##
## data:  model
## BP = 6.4939, df = 1, p-value = 0.01082
```

- Breusch-Pagan test: a regression is made, where the dependent variable consists of the squared residuals as to assess whether the independent variables of the model have any relationship with the `residuals`. We expect the effect to be 0 because if the error variance is constant, the error should vary in relation to the values of the x
- The p-value is less than 0.05, the null hypothesis is rejected. We have some issues with heteroscedasticity

Solution to heteroscedasticity

- Solution, Robust Standard Error

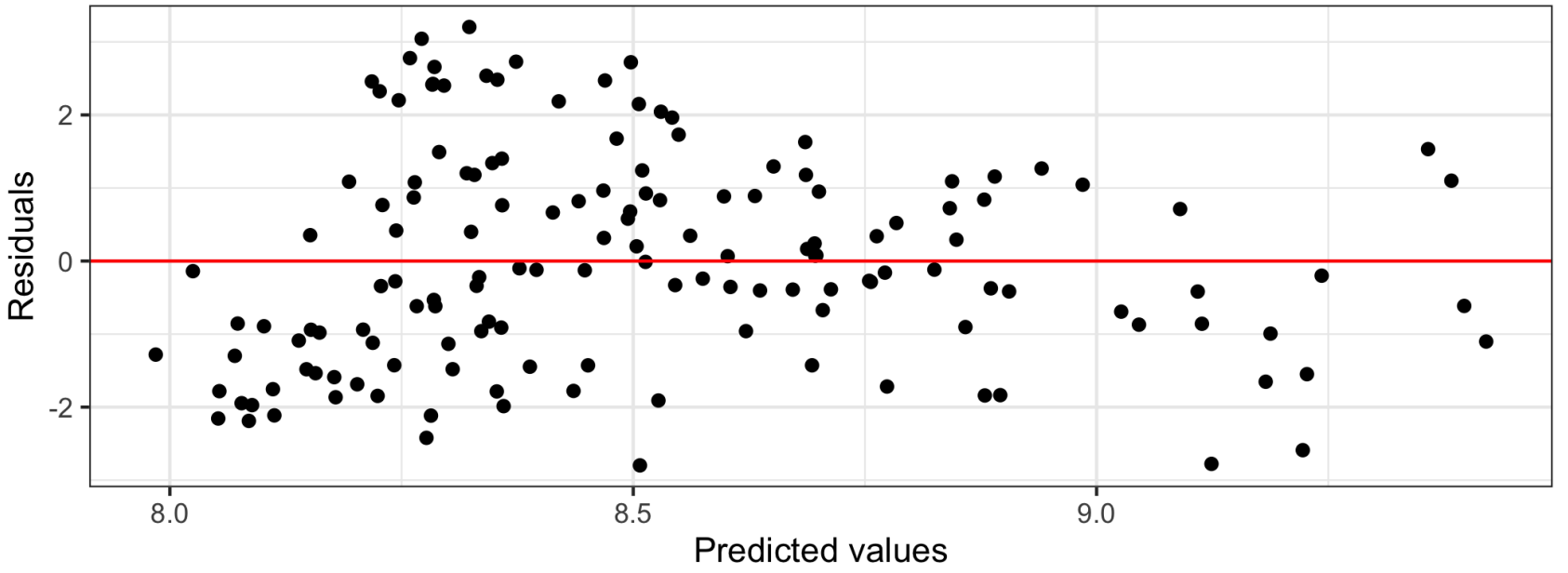
```
library(sandwich)
coeftest(model, vcov = vcovHC(model, "HC3") )
```

```
##
## t test of coefficients:
##
##              Estimate Std. Error t value  Pr(>|t|)
## (Intercept)  7.98275    0.24709 32.3074 < 2.2e-16
## miliper      0.79971    0.28800  2.7768  0.006211
```

- HC3 (highly recommended); HC1(the stata software version)

Checking linearty

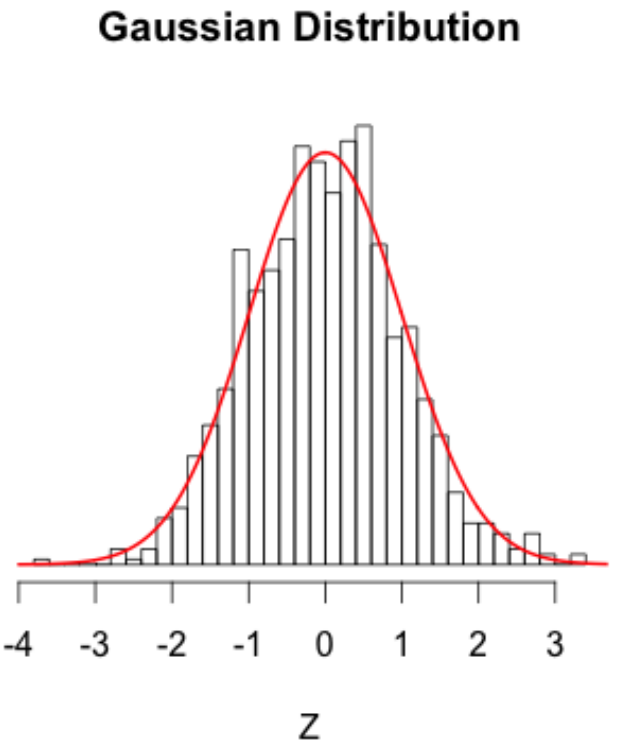
```
df <- df %>% mutate(fitted.values=fitted(model))
ggplot(data = df, aes(x =fitted.values, y = residuals)) +
  geom_point() +
  geom_hline(yintercept=0,color="red")+
  labs(x = "Predicted values", y = "Residuals")
```

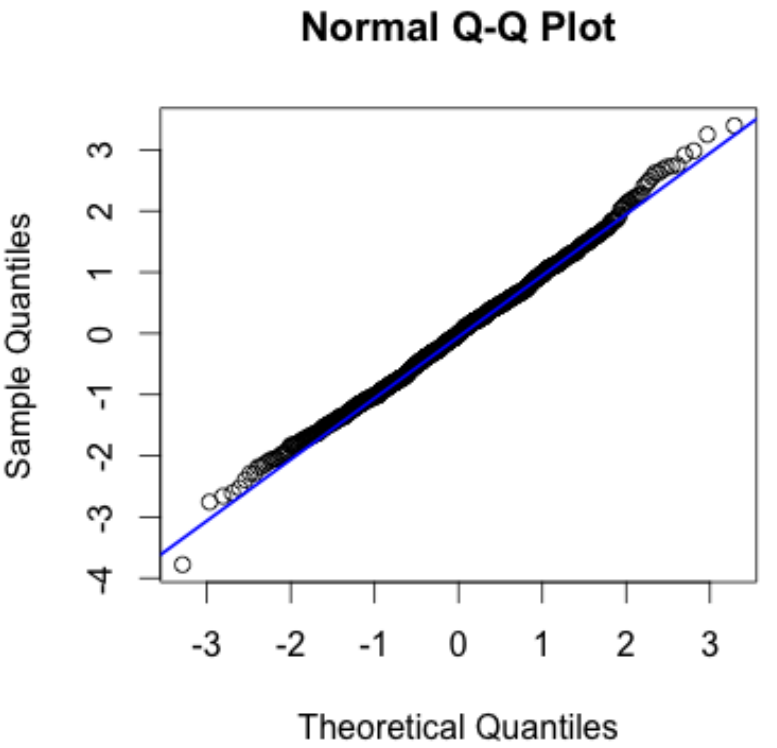


Checking Normality

- Examine the distribution of the residuals to determine if the Normality assumption is satisfied
- Plot the residuals in a histogram and a Normal QQ plot to visualize their distribution and assess Normality
- Most inference methods for regression are robust to some departures from Normality

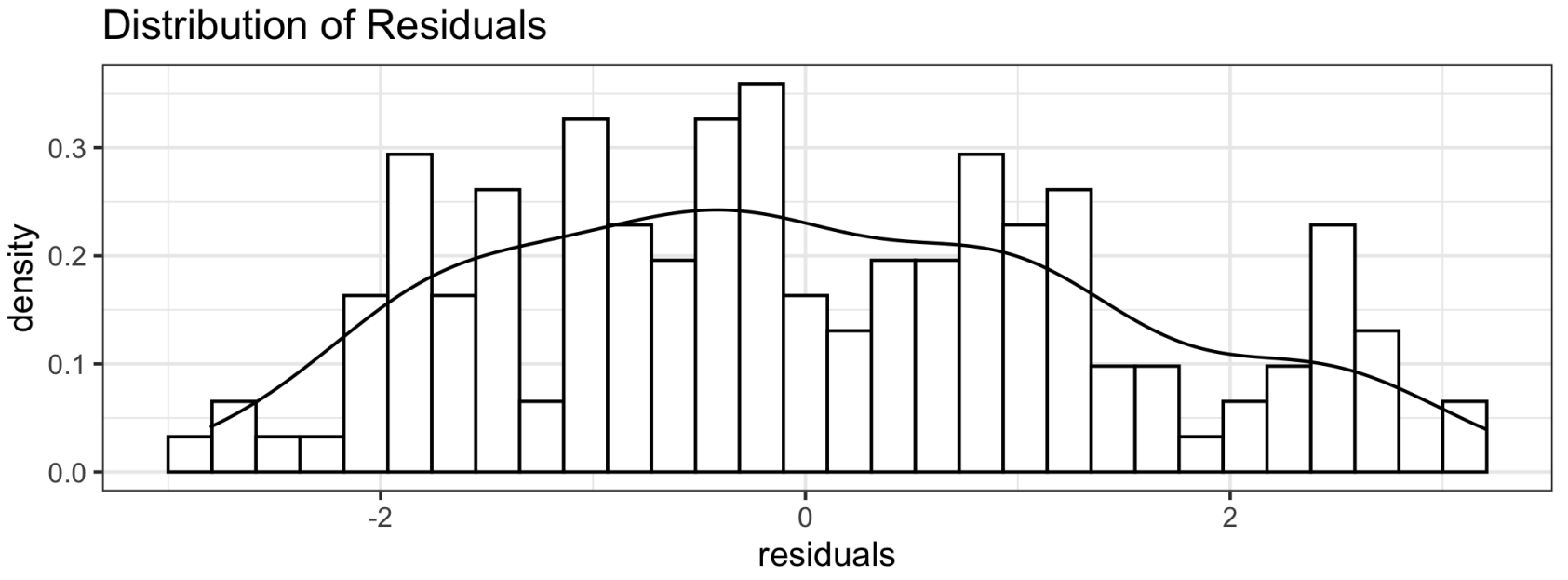
Normal QQ-Plot



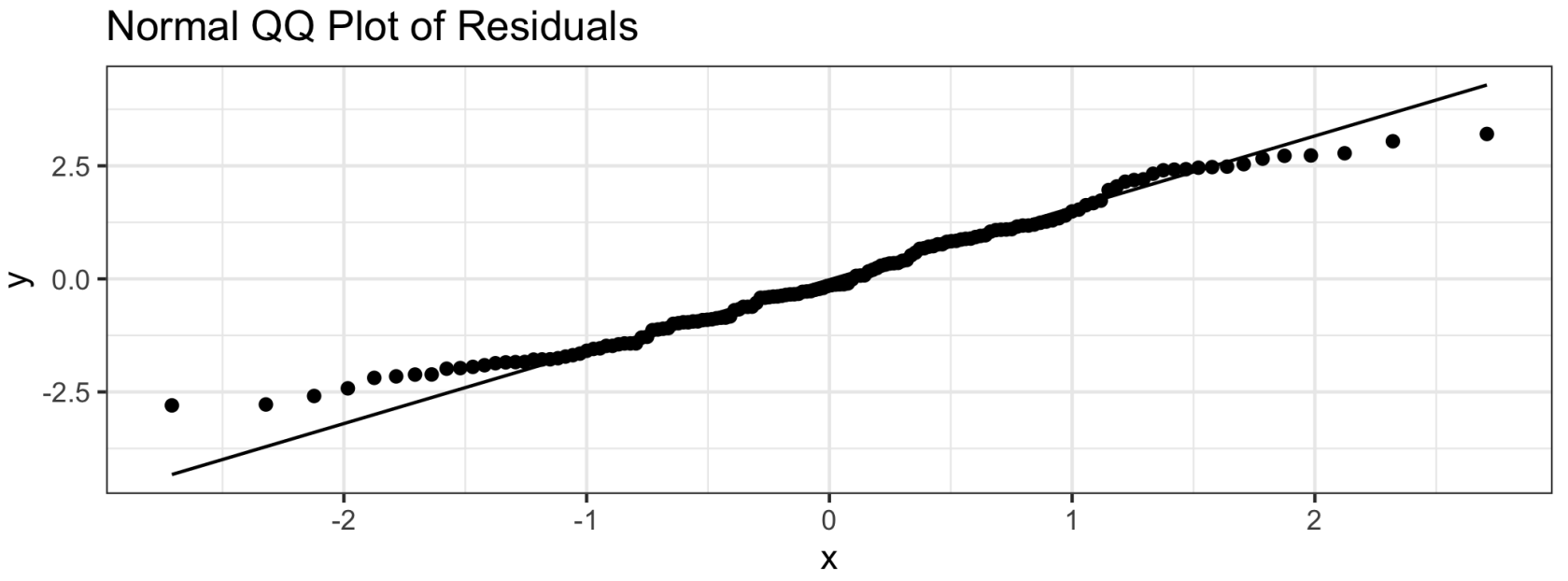


Checking normality with R

```
ggplot(data=df,aes(x=residuals)) +
  geom_histogram(aes(y=..density..), colour="black", fill="white") + geom_dens
  labs(title="Distribution of Residuals")
```

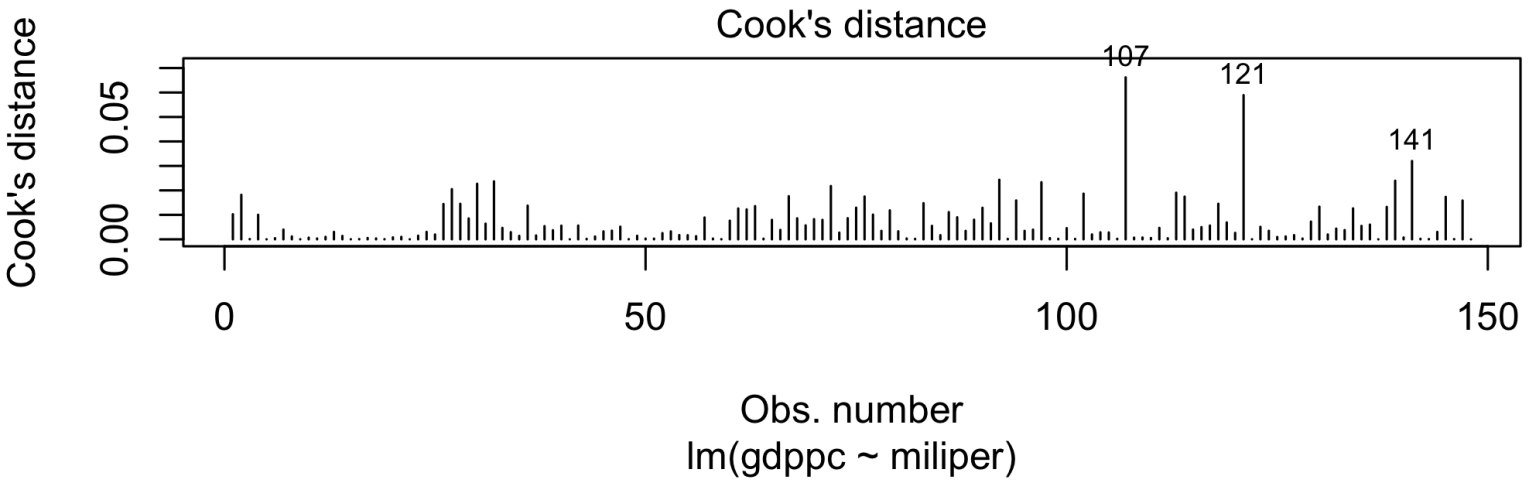


```
ggplot(data=df, aes(sample=residuals)) +
  stat_qq() +
  stat_qq_line() +
  labs(title="Normal QQ Plot of Residuals")
```



Checking influential values

```
plot(model, which = 4, id.n = 3)
```



- not all outliers are influential observations.

```r
# Extract model results
augment(model) %>%
  mutate(index = 1:n())%>%
  top_n(3, .cooksd)
```

```
## # A tibble: 3 × 9
##   gdppc miliper .fitted .resid   .hat .sigma .cooksd .std.resid index
##   <dbl>   <dbl>   <dbl>  <dbl>  <dbl>  <dbl>   <dbl>      <dbl> <int>
## 1  6.63    1.55    9.22  -2.59 0.0380   1.43  0.0662      -1.83   107
## 2  6.35    1.43    9.12  -2.78 0.0299   1.43  0.0589      -1.95   121
## 3 10.9     1.72    9.36   1.53 0.0511   1.44  0.0320       1.09   141
```

Checking Independence

- Often, we can conclude that the independence assumption is sufficiently met based on a description of the data and how it was collected.
- Two common violations of the independence assumption:
  - Serial Effect: If the data were collected over time, the residuals should be plotted in time order to determine if there is serial correlation
  - Cluster Effect: You can plot the residuals vs. a group identifier or use different markers (colors/shapes) in the residual plot to determine if there is a cluster effect.

Multicollinearity

Multicollinearity corresponds to a situation where the data contain highly correlated predictor variables.

```
model2 <- lm(gdppc ~ miliper + dem + vio, data = df)
car::vif(model2)
```

```
##  miliper      dem      vio
## 1.055984 1.067529 1.031670
```

- As a rule of thumb, a VIF (variance inflation factors) value that exceeds 5 indicates a problematic amount of collinearity.
- Solution: 1) remove one of the independent variables that is strongly correlated; 2) combine the variables that are strongly correlated.