PS 1

ARE 212, Spring 2026

David Cai, Miriam Gold, Ella Moxley, Geoffrey Yip

2026-02-25

Set-up

```
library(WDI)
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.2     v tibble    3.2.1
v lubridate 1.9.4     v tidyr     1.3.1
v purrr     1.0.4
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
```

```
library(magrittr)
```

```
Attaching package: 'magrittr'

The following object is masked from 'package:purrr':

    set_names

The following object is masked from 'package:tidyr':

    extract
```

```r
library(fs)
library(janitor)
```

```
Attaching package: 'janitor'

The following objects are masked from 'package:stats':

    chisq.test, fisher.test
```

```r
# Paths ==================
path <- "/Users/miriamgold/projects/ARE212_2026/ps1"
path_functions <- file.path(path, "functions")
path_data <- file.path(path, "data")

# Source custom functions
dir_walk(path_functions, source)

# Data setup ==============

iso_codes <-
  read_csv("https://gist.githubusercontent.com/radcliff/f09c0f88344a7fcef373/raw/2753c482ad0
```

```
Rows: 246 Columns: 5
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (5): English short name lower case, Alpha-2 code, Alpha-3 code, Numeric ...

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
wb_vars <- c("EN.GHG.CO2.MT.CE.AR5", "NY.GDP.MKTP.KD", "SP.POP.TOTL")

wdi_avail_countries <-
  WDI::WDI_data %>%
  purrr::pluck(2) %>%
  pull(iso2c)

countries <-
  iso_codes |>
```

```
  filter(`Alpha-2 code` %in% wdi_avail_countries) %>%
  pull("Alpha-2 code")
```

Q1

```
# I had to disable the automatic data fetching because the WB API servers stopped responding
if(FALSE){
wdi_data_raw <-
  countries %>%
  purrr::map(~wdi_noisy(country = .x, indicator=wb_vars, start=2010, end=2010)) %>%
  purrr::list_rbind()
}

wdi_data_raw <-
  read_csv(
    file.path(path_data, "48038096-3fbd-47af-8962-7f774cfef8a8_Data.csv")
  )
```

```
Rows: 656 Columns: 5
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (5): Country Name, Country Code, Series Name, Series Code, 2010 [YR2010]

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Q2

```
wdi_data_clean <-
  wdi_data_raw |>
  clean_names() %>%
  drop_na() %>%
  select(country_code, series_code, x2010_yr2010) %>%
  mutate(
    var_name = case_when(
      series_code == "EN.GHG.CO2.MT.CE.AR5" ~ "CO2",
      series_code == "NY.GDP.MKTP.KD" ~ "GDP",
```

```
      series_code == "SP.POP.TOTL" ~ "POP"
    ),
    x2010_yr2010 = as.numeric(x2010_yr2010)
  ) |>
    rename(country = country_code) %>%
    pivot_wider(
      id_cols = country,
      names_from = "var_name",
      values_from = "x2010_yr2010"
    ) %>%
  drop_na()
```

```
Warning: There was 1 warning in `mutate()`.
i In argument: `x2010_yr2010 = as.numeric(x2010_yr2010)`.
Caused by warning:
! NAs introduced by coercion
```

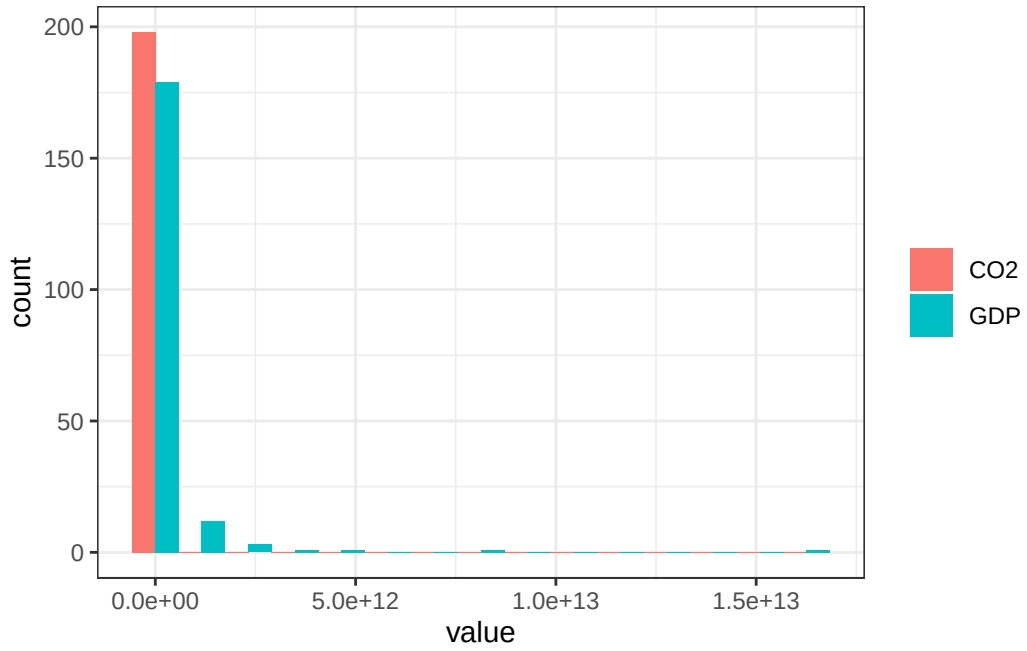
Q3

```
wdi_summary_stats <-
  wdi_data_clean |>
   pivot_longer(cols = c("CO2", "GDP", "POP"), names_to = "var") |>
   group_by(var) |>
   summarise(
    mean = mean(value),
    sd = sd(value),
    min = min(value),
    max = max(value)
   )
print(wdi_summary_stats)
```

```
# A tibble: 3 x 5
  var            mean      sd       min     max
  <chr>         <dbl>   <dbl>     <dbl>   <dbl>
1 CO2            164. 7.81e 2        0  9.11e 3
2 GDP   325737923786. 1.36e12 30560853. 1.63e13
3 POP       34989968. 1.34e 8    10043  1.34e 9
```

Q4

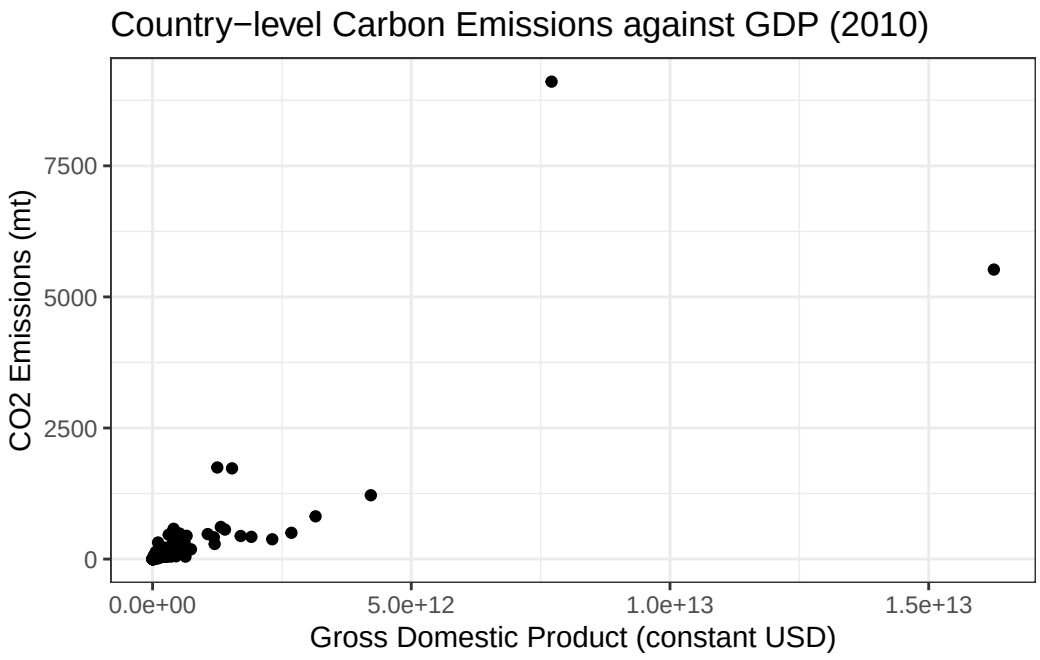
```
wdi_data_clean %>%
   pivot_longer(cols = c("CO2", "GDP", "POP"), names_to = "var") %>%
  filter(var %in% c("CO2", "GDP")) %>%
  ggplot(aes(x = value, fill = var)) +
  geom_histogram(bins = 15, position = "dodge") +
  scale_fill_discrete(name = NULL) +
  theme_bw()
```



Q5

```
wdi_data_clean %>%
  ggplot(aes(x = GDP, y = CO2)) +
  geom_point() +
  scale_x_continuous("Gross Domestic Product (constant USD)") +
  scale_y_continuous("CO2 Emissions (mt)") +
  labs(
    title = "Country-level Carbon Emissions against GDP (2010)"
```

```
  ) +
  theme_bw()
```



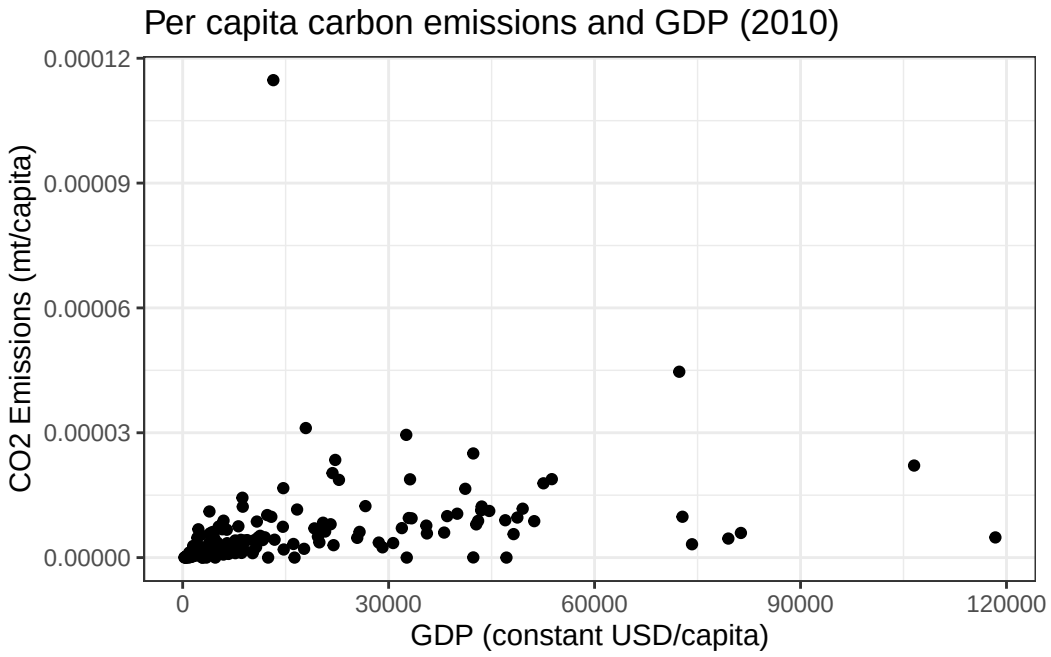
Q6/7

```
wdi_per_capita <-
  wdi_data_clean %>%
  mutate(
    CO2pc = CO2/POP,
    GDPpc = GDP/POP
  )
```

Q8

```
wdi_per_capita %>%
  ggplot(aes(x = GDPpc, y = CO2pc)) +
  geom_point() +
```

```
  scale_x_continuous("GDP (constant USD/capita)") +
  scale_y_continuous("CO2 Emissions (mt/capita)") +
  labs(
    title = "Per capita carbon emissions and GDP (2010)"
  ) +
  theme_bw()
```

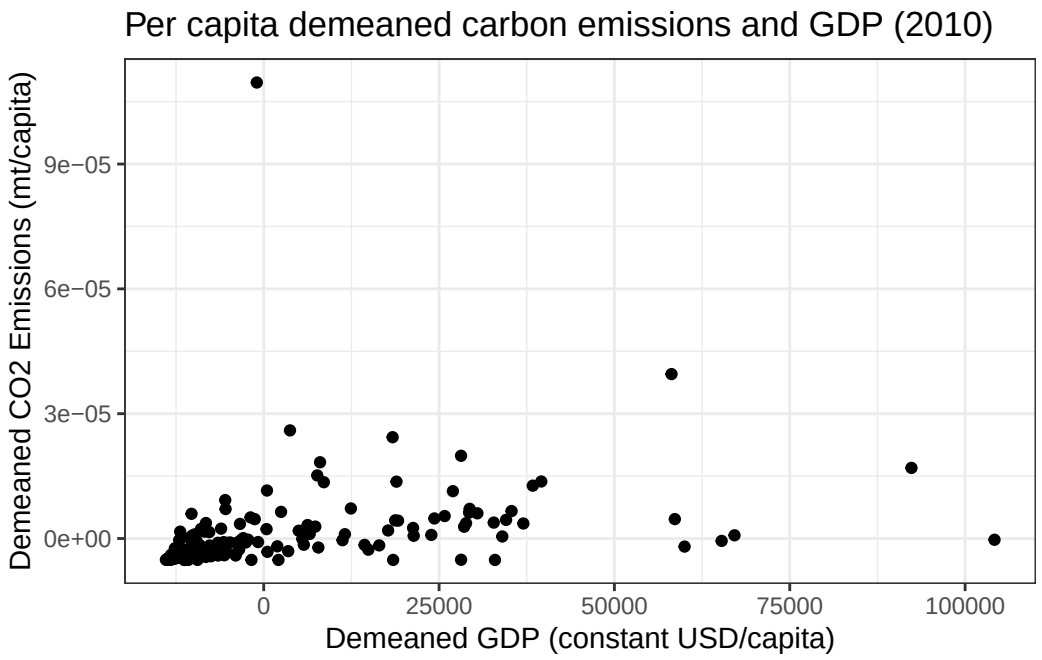


Q9

```
wdi_per_capita_demeaned <-
  wdi_per_capita %>%
  mutate(
    CO2pcdev = CO2pc-mean(CO2pc),
    GDPpcdev = GDPpc-mean(GDPpc)
  )
```

Q10

```
wdi_per_capita_demeaned %>%
  ggplot(aes(x = GDPpcdev, y = CO2pcdev)) +
  geom_point() +
  scale_x_continuous("Demeaned GDP (constant USD/capita)") +
  scale_y_continuous("Demeaned CO2 Emissions (mt/capita)") +
  labs(
    title = "Per capita demeaned carbon emissions and GDP (2010)"
  ) +
  theme_bw()
```



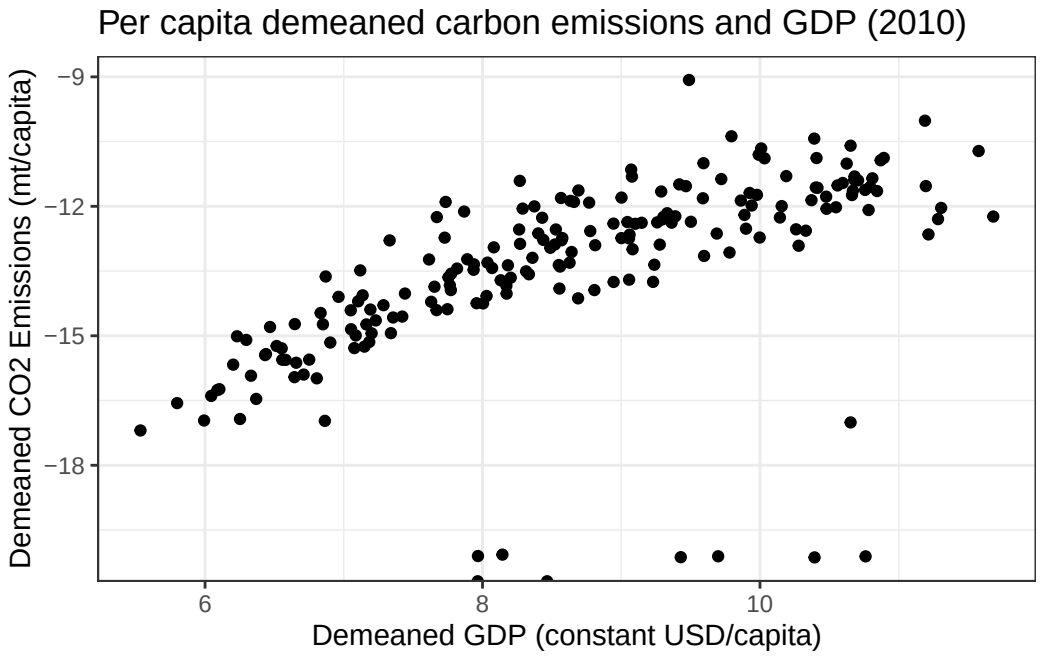
Q11

```
wdi_per_capita_log <-
  wdi_per_capita_demeaned %>%
  mutate(
    CO2pcln = log(CO2pc),
```

```
    GDPpcln = log(GDPpc)
  )
```

Q12

```
wdi_per_capita_log %>%
  ggplot(aes(x = GDPpcln, y = CO2pcln)) +
  geom_point() +
  scale_x_continuous("Demeaned GDP (constant USD/capita)") +
  scale_y_continuous("Demeaned CO2 Emissions (mt/capita)") +
  labs(
    title = "Per capita demeaned carbon emissions and GDP (2010)"
  ) +
  theme_bw()
```



Q13

```
  wdi_per_capita_log %>%
    write_csv(file = file.path(path_data, "are212_gold_ps1_wdi_clean.csv"))
```

Q14

We define a custom OLS regression function below

```
ols <- function(x, y, intercept = FALSE) {

  X <- as.matrix(x)
  Y <- as.matrix(y)

  if(intercept) {
    intercept_vector <- matrix(1, nrow(X))
    X <- cbind(intercept_vector, X)
  }


  N <- nrow(X)
  k <- ncol(X)
  b_ols <- solve(t(X)%*%X) %*% (t(X)%*%Y) # b = (X'X)^{-1}(X'Y)
  y_hat <- (X) %*% b_ols

  r2_uc <- (t(y_hat)%*%y_hat) / (t(Y)%*%Y)
  r2 <- t(y_hat-mean(Y))%*%(y_hat-mean(Y)) / t(Y-mean(Y))%*%(Y-mean(Y))
  r2 <- r2[[1,1]]
  r2_bar <- 1 - ((N-1)/(N-k))*(1-r2)

  e <- Y-y_hat
  s2 <- (t(e)%*%e)/(N-k)

  list(
    beta = b_ols,
    X = X,
    Y = Y,
    N = N,
    k = k,
    df = N-k,
    r2_uc = r2_uc,
    r2 = r2,
```

```
    r2_bar = r2_bar,
    y_hat = y_hat,
    e = e,
    s2 = s2
  )
}
```

Next, run each specification and report OLS coefficients

```
ols(
  wdi_per_capita_log$GDPpc,
  wdi_per_capita_log$CO2pc
)$beta
```

```
             [,1]
[1,] 2.374949e-10
```

```
ols(
  wdi_per_capita_log$GDPpc,
  wdi_per_capita_log$CO2pc*1000
)$beta # the coefficient has been multiplied by 1000)
```

```
             [,1]
[1,] 2.374949e-07
```

```
ols_co2_gdp <-
  ols(
    wdi_per_capita_log$GDPpc/1000,
    wdi_per_capita_log$CO2pc*1000
  )
ols_co2_gdp$beta
```

```
             [,1]
[1,] 0.0002374949
```

In the last regression, the coefficient is multiplied by a further factor of 1000

Q15

```r
## N = 197
n <- nrow(wdi_per_capita_log)

## Degrees of freedom: N-k
df <- n-1 # k=1 because there is no intercept

## beta
b <- ols_co2_gdp$beta

## R^2 uncentered
ols_co2_gdp$r2_uc
```

```
          [,1]
[1,] 0.2636317
```

```r
## R^2
ols_co2_gdp$r2
```

```
[1] 0.2513099
```

```r
## Adjusted R^2
ols_co2_gdp$r2_bar
```

```
[1] 0.2513099
```

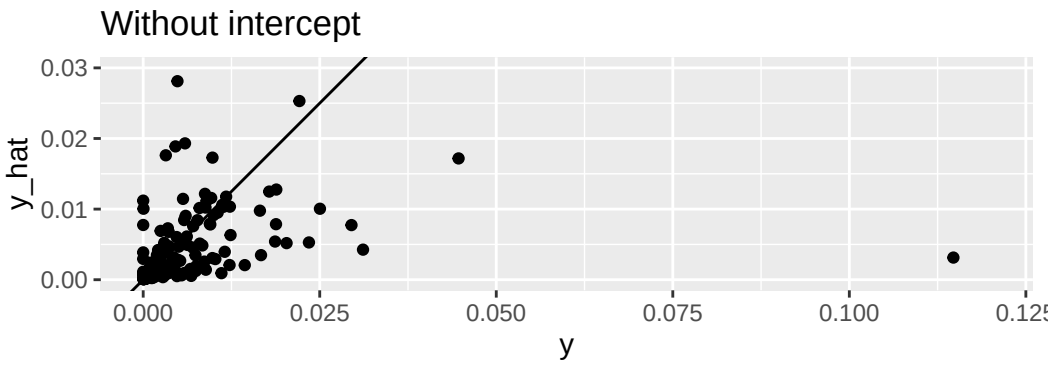
```r
## s^2
ols_co2_gdp$s2
```

```
             [,1]
[1,] 9.410012e-05
```

```r
predicted_vs_actual <-
  data.frame(
    y_hat = ols_co2_gdp$y_hat,
    y = ols_co2_gdp$Y
)

predicted_vs_actual %>%
```

```
  ggplot(aes(x = y, y = y_hat)) +
  geom_point() +
  geom_abline() +
  coord_fixed() +
  scale_y_continuous(limits = c(0, 0.03)) +
  scale_x_continuous(limits = c(0, 0.12)) +
  labs(
    title = "Without intercept"
  )
```



Q16

```
ols_co2_gdp_intercept <-
  ols(
    wdi_per_capita_log$GDPpc/1000,
    wdi_per_capita_log$CO2pc*1000,
    intercept = TRUE
  )

## beta
```

```r
b_i <- ols_co2_gdp_intercept$beta

## R^2 uncentered
ols_co2_gdp_intercept$r2_uc
```

```
          [,1]
[1,] 0.3010531
```

```r
## R^2
ols_co2_gdp_intercept$r2
```

```
[1] 0.1173416
```

```r
## Adjusted R^2
ols_co2_gdp_intercept$r2_bar
```

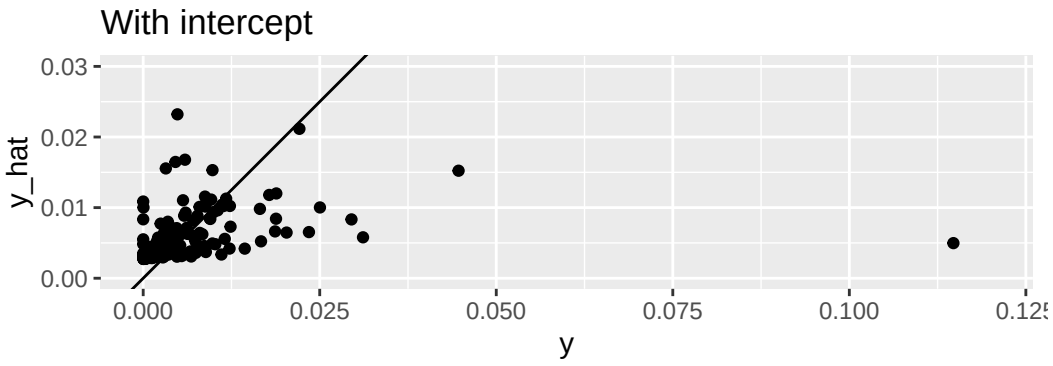
```
[1] 0.1128383
```

```r
## s^2
ols_co2_gdp_intercept$s2
```

```
             [,1]
[1,] 8.977376e-05
```

```r
predicted_vs_actual_intercept <-
  data.frame(
    y_hat = ols_co2_gdp_intercept$y_hat,
    y = ols_co2_gdp_intercept$Y,
    x = ols_co2_gdp_intercept$X[,2]
)

predicted_vs_actual_intercept %>%
  ggplot(aes(x = y, y = y_hat)) +
  geom_point() +
  geom_abline() +
  coord_fixed() +
  scale_y_continuous(limits = c(0, 0.03)) +
  scale_x_continuous(limits = c(0, 0.12)) +
  labs(
    title = "With intercept"
  )
```



Q17

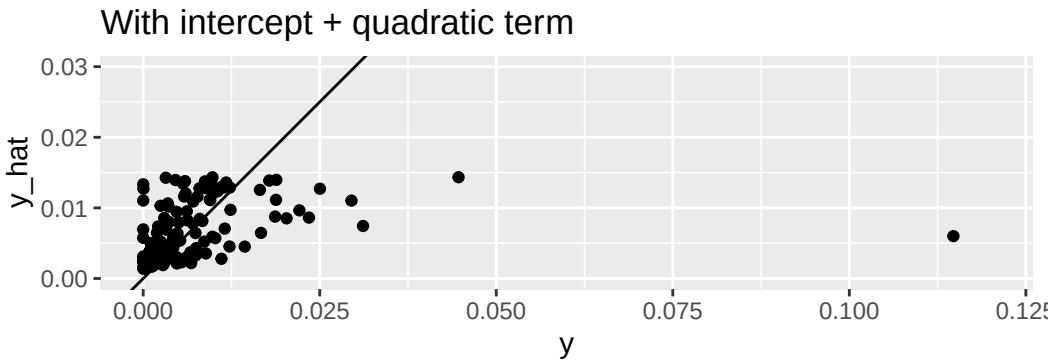
```
wdi_per_capita_quad <-
  wdi_per_capita_log %>%
  mutate(
    CO2pc_thou = CO2pc*1000,
    GDPpc_thou = GDPpc/1000,
    GDPpc_thou_2 = (GDPpc_thou^2)
  )

ols_co2_gdp_quad <-
  ols(
    x = wdi_per_capita_quad %>% select(GDPpc_thou, GDPpc_thou_2),
    y = wdi_per_capita_quad %>% select(CO2pc_thou),
    intercept = TRUE
  )

predicted_vs_actual_quad <-
  data.frame(
    y_hat = ols_co2_gdp_quad$y_hat,
```

```
    y = ols_co2_gdp_quad$Y,
    x = ols_co2_gdp_quad$X[,2]
) %>%
  rename(y_hat = 1, y = 2, x = 3)

predicted_vs_actual_quad %>%
  ggplot(aes(x = y, y = y_hat)) +
  geom_point() +
  geom_abline() +
  coord_fixed() +
  scale_y_continuous(limits = c(0, 0.03)) +
  scale_x_continuous(limits = c(0, 0.12)) +
  labs(
    title = "With intercept + quadratic term"
  )
```



In this regression, we now add a quadratic term for GDP per capita. This does not violate the first two assumptions. We are still linear in parameters; it's just that we're linear in a squared term now too. We still have full rank because the quadratic term is not a simple rescaling of our original GDP per capita term.

We may have violated spherical errors, though it's hard to tell for sure.

The specification does make economic sense. This is the Kuznets Curve story, where firms adopt industrize as they develop but then develop cleaner technology once they get rich enough.

Q18

```
wdi_per_capita_demean <-
  wdi_per_capita_quad %>%
  mutate(
    CO2pc_thou_demean = CO2pc_thou - mean(CO2pc_thou),
    GDPpc_thou_demean = GDPpc_thou - mean(GDPpc_thou),
    GDPpc_thou_2_demean = GDPpc_thou_2 - mean(GDPpc_thou_2)
  )

ols_co2_gdp_demean <-
  ols(
    x = wdi_per_capita_demean %>% select(GDPpc_thou_demean, GDPpc_thou_2_demean),
    y = wdi_per_capita_demean %>% select(CO2pc_thou_demean),
    intercept = FALSE
  )

ols_co2_gdp_quad$beta
```

```
               CO2pc_thou
              1.295359e-03
GDPpc_thou    3.959382e-04
GDPpc_thou_2 -2.979636e-06
```

```
ols_co2_gdp_demean$beta
```

```
                    CO2pc_thou_demean
GDPpc_thou_demean         3.959382e-04
GDPpc_thou_2_demean      -2.979636e-06
```

The beta's we get from a regression where we demean the variables will be about the same to the results we get from question 17. Even though we do not include an explicit intercept, the demeaning allows us to recover the results from question 17 because we are centering the data at zero and keeping the slope the same.

Q19

```
ols_co2_gdp_q19 <-
  ols(
    x = wdi_per_capita_quad %>% select(GDPpc_thou),
    y = wdi_per_capita_quad %>% select(CO2pc_thou),
    intercept = TRUE
  )

resid_q19 <- ols_co2_gdp_q19$e

ols_1_gdp <-
  ols(
    x = wdi_per_capita_quad %>% select(GDPpc_thou),
    y = matrix(1, nrow = nrow(wdi_per_capita_quad)),
    intercept = FALSE
  )

ols_1_gdp_resid <- ols_1_gdp$e


ols_gdp2_gdp <-
  ols(
    x = wdi_per_capita_quad %>% select(GDPpc_thou),
    y = wdi_per_capita_quad %>% select(GDPpc_thou_2),
    intercept = FALSE
  )

ols_gdp2_gdp_resid <- ols_gdp2_gdp$e

resid_matrix <- cbind(ols_1_gdp_resid, ols_gdp2_gdp_resid)

ols_resid_on_resid <-
  ols(
    x = resid_matrix,
    y = resid_q19,
    intercept = FALSE
  )

ols_resid_on_resid$beta
```

```
                CO2pc_thou
              -1.387507e-03
GDPpc_thou_2 -2.979636e-06
```

In question 17, we regressed CO2pc = b1 + b2*GDPpc + b3*GDPpc^2. In question 19, we regressed CO2pc on GDPpc, and obtained the residual e1 1 on GDPpc, and obtained the residual e2 GDPpc^2 on GDPpc, and obtained the residual e3 Finally we regressed e1 = a1*e2 + a2*e3. we see saw the coefficient on e2 is the same as b1; the coefficient on e3 is the same as b3. That is, a1 = b1 and a2 = b3. This is an another example of FWL Thereom in action. Removing the variation in the CO2pc, 1, and GDPpc^2 that comes from GDPpc and using the residual variation as regressand/regressors gives you the same coefficients as the original equation in Q17.