

Preprocesamiento de datos

true

Contents

Inicio	2
Variables	2
Producción	2
Precio del mercado interno	2
Precios de Referencia del Mercado Mundial	2
Comercio exterior	4
Retenciones	5
Regalías	5
Subsidios	5
Empleo, remuneraciones y masa Salarial	6
Activos	6
Valor total de la producción	11
Tasa de ganancia y renta apropiada por las empresas	13
Rentabilidad total de la rama	14
Renta apropiada por las empresas de la rama	14
Renta por el diferencial de precios entre el mercado interno y las referencias internacionales	14
Renta apropiada por sobrevaluación cambiaria	14
Renta apropiada por el Estado mediante impuestos específicos	15
Renta Hidrocarburífera Total. En precios constantes, sobre Plusvalía Total y PBI	15
Método directo (a partir de descuentos sobre el VBP)	15
Método indirecto (suma de mecanismos)	15
Costos	16
Comparación con estimación de otros autores	17

Inicio

Variables

Producción

Petróleo crudo

Gas natural

Produccion total, en barriles equivalentes de petróleo

```
produccion_total <- prod_merge_gas_bep %>%
  select(anio, unidad, prod_gas) %>%
  left_join(prod_merge_crudo %>%
    select(anio, unidad, prod_crudo) %>% mutate(unidad = "BEP") ) %>%
  mutate(produccion_total_bep = prod_crudo + prod_gas)
```

```
## Joining, by = c("anio", "unidad")
```

Precio del mercado interno

Petróleo crudo

Gas natural

Precios de Referencia del Mercado Mundial

Precios de Referencia del Crudo

```
## # A tibble: 58 x 8
##   anio unidad      precio_expo_comt~ precio_expo_rega~ precio_expo_mec~ ...1
##   <dbl> <chr>          <dbl>          <dbl>          <dbl> <dbl>
## 1 1962 USD/barriles      2.31            NA            NA      NA
## 2 1963 USD/barriles      2.18            NA            NA      NA
## 3 1964 USD/barriles      1.97            NA            NA      NA
## 4 1965 USD/barriles      NA              NA            NA      NA
## 5 1966 USD/barriles      5.51            NA            NA      NA
## 6 1967 USD/barriles      1.78            NA            NA      NA
## 7 1968 USD/barriles      1.30            NA            NA      NA
## 8 1969 USD/barriles      2.35            NA            NA      NA
## 9 1970 USD/barriles      NA              NA            NA      NA
## 10 1971 USD/barriles      NA              NA            NA      NA
## # ... with 48 more rows, and 2 more variables: precio_expo_crudo_indec <dbl>,
## #   precio_expo_comtrade_hs_crudo <dbl>
```

Precios de Referencia del Gas

Exportación Argentina

Referencias mundiales

Bolivia

```
precio_crudo_vs_gas <- precio_mdomundial_gas_bep %>%
  select(anio, unidad, precio_externo_gas) %>%
  left_join(precios_referencia_y_expo_crudo %>%
    select(-unidad) %>%
    gather(key = tipo_precio_crudo,
           value = precio_crudo_usd_bbl,
           2:ncol()), by = "anio") %>%
  mutate(precio_crudo_sobre_gas = precio_crudo_usd_bbl/precio_externo_gas) %>%
  filter(!(is.na(precio_crudo_sobre_gas)),
         !(tipo_precio_crudo %in% c("wti_spot_price_fred", "precio_me_crudo"))) )
  # (tipo_precio_crudo %in% c("brent_iea")) )
  # (tipo_precio_crudo %in% c("brent_iea", "brent_historic")) )
precio_crudo_vs_gas
```

Todos los precios

```
## # A tibble: 215 x 6
## # Groups:   anio [51]
##   anio unidad  precio_externo_gas tipo_precio_crudo  precio_crudo_usd_~
##   <dbl> <chr>          <dbl> <chr>                <dbl>
## 1 1964 USD/bep      0.634 precio_expo_comtrade_sit~ 1.97
## 2 1964 USD/bep      0.634 brent_historic           3
## 3 1965 USD/bep      0.521 brent_historic          3.01
## 4 1966 USD/bep      0.228 precio_expo_comtrade_sit~ 5.51
## 5 1966 USD/bep      0.228 brent_historic           3.1
## 6 1967 USD/bep      0.228 precio_expo_comtrade_sit~ 1.78
## 7 1967 USD/bep      0.228 brent_historic           3.12
## 8 1972 USD/bep      0.000893 brent_historic           3.6
## 9 1973 USD/bep      6.05 brent_historic           4.75
## 10 1974 USD/bep      1.72 precio_expo_comtrade_sit~ 15.2
## # ... with 205 more rows, and 1 more variable: precio_crudo_sobre_gas <dbl>
```

```
# graf <- precio_crudo_vs_gas %>%
#   ggplot(aes(anio, precio_crudo_sobre_gas, color=tipo_precio_crudo))+
#   geom_line()+
#   labs(title = "precio del crudo sobre el del gas (ambos en barriles)")

# ggplotly(graf, width = 800, length=600)
# graf
```

Comercio exterior

Exportaciones de crudo

```
impo_crudo = comex_crudo_sesco %>%
  filter(variable == "importacion", unidad != "ton") %>%
  mutate(fuente = "Sec. Energía - SESCO",
         variable = "Importación de petróleo crudo") %>%
  bind_rows(data_comtrade %>%
    filter(trade_flow == "Import",
           commodity_code == 33101) %>%
    mutate( producto = "crudo",
           variable = "Importación de petróleo crudo", fuente = "UN-Comtrade") %>%
    select( anio, variable, producto,
           USD = valor , barriles = cantidad, fuente) %>%
    pivot_longer(cols = c(USD,barriles), names_to = "unidad",
                 values_to = "valor", names_repair = "unique")) %>%
  bind_rows(comex_hidrocarburos_meccon %>%
    select(anio, valor = impo_bbl_crudo) %>%
    mutate( fuente = "MECON",
           variable = "Importación de petróleo crudo", producto = "crudo", unidad = "barriles" ) )
```

Importaciones de crudo

Exportaciones de gas natural

```
impo_gas = comex_gas %>%
  filter(variable == "importacion", unidad != "ton") %>%
  mutate(fuente = "Sec. Energía - SESCO",
         variable = "Importación de gas natural") %>%
  bind_rows(data_comtrade %>%
    filter(trade_flow == "Import",
           commodity_code == 3411) %>%
    mutate( producto = "gas",
           variable = "Importación de gas natural",
           fuente = "UN-Comtrade") %>%
    select( anio, variable, producto,
           USD = valor , m3 = cantidad, fuente) %>%
    pivot_longer(cols = c(USD,m3), names_to = "unidad",
                 values_to = "valor", names_repair = "unique")) %>%
  bind_rows(comex_hidrocarburos_meccon %>%
    select(anio, valor = impo_MM3_gas) %>%
    mutate( fuente = "MECON", valor = valor*1e6,
           variable = "Importación de gas natural", producto = "gas", unidad = "m3" ) )
```

Importaciones de gas natural

Retenciones

```
## Joining, by = "anio"
```

```
## Joining, by = c("anio", "unidad")
```

Regalías

Subsidios

```
# CEFIP cont et al. Subsidios como % del PBI
subsidios_cefip <- read_excel("../data/cefip/subsidios.xlsx") %>%
  gather(key = anio,
         value = subsidios_porcentaje_pbi,
         2:ncol()) %>%
  filter(sector %in% c( "Plan Gas",
                      # "ENARSA", "CAMMESA",
                      "Subsidios FF GN y GLP")) %>%
  mutate(anio = as.double(anio)) %>%
  left_join(ganancia_y_pbi %>%
            select(anio, pbi)) %>%
  mutate(subsidios_porcentaje_pbi=as.double(subsidios_porcentaje_pbi),
         subsidios_cefip = subsidios_porcentaje_pbi/100 * pbi * 10^6) %>%
  group_by(anio) %>%
  summarise(unidad = "Pesos corrientes",
            subsidios_cefip=sum(subsidios_cefip, na.rm = T))
```

```
## Joining, by = "anio"
```

```
## Warning in mask$eval_all_mutate(quo): NAs introducidos por coerción
```

```
#ejes
subsidios_ejes <- read_excel("../data/ejes/subsidios.xlsx" ) %>%
  rename(subsidios_usd = subsidios_hidrocarburos) %>%
  right_join(tcp_anual_b %>%
            select(anio, tcc), by = "anio") %>%
  group_by(anio) %>%
  summarise(unidad = "Pesos corrientes",
            subsidios_ejes = subsidios_usd*tcc*10^6)

subsidios_hidrocarburos <- subsidios_ejes%>%
  select(anio, unidad, subsidios_ejes) %>%
  full_join(subsidios_cefip, by =c("anio", "unidad")) %>%
  arrange(-anio) %>%
  mutate(unidad = "Pesos corrientes")
  # x = subsidios_cefip/subsidios_ejes)
subsidios_hidrocarburos
```

```
## # A tibble: 84 x 4
```

```
##   anio  unidad      subsidios_ejes subsidios_cefip
```

```
##      <dbl> <chr>                                <dbl>          <dbl>
## 1 2019 Pesos corrientes                        NA           6434174958.
## 2 2018 Pesos corrientes                        NA           29085444296.
## 3 2017 Pesos corrientes                        NA           21320456990.
## 4 2016 Pesos corrientes      11821956951.      22216030803.
## 5 2015 Pesos corrientes      27931660000      35727065374.
## 6 2014 Pesos corrientes      18139998800      11447716064.
## 7 2013 Pesos corrientes      11721150000      10044925465.
## 8 2012 Pesos corrientes      4343102667.      6330993236.
## 9 2011 Pesos corrientes      5478630000      1089512052.
## 10 2010 Pesos corrientes      3967800000      498516278.
## # ... with 74 more rows
```

Empleo, remuneraciones y masa Salarial

```
## # A tibble: 32 x 7
##   anio unidad      masa_salarial_extr~ masa_salarial_tot~ masa_salarial_extr~
##   <dbl> <chr>          <dbl>          <dbl>          <dbl>
## 1 1980 Millones de~      NA              NA              NA
## 2 1981 Millones de~      NA              NA              NA
## 3 1982 Millones de~      NA              NA              NA
## 4 1983 Millones de~      NA              NA              NA
## 5 1984 Millones de~      NA              NA              NA
## 6 1985 Millones de~      NA              NA              NA
## 7 1986 Millones de~      NA              NA              NA
## 8 1987 Millones de~      NA              NA              NA
## 9 1996 Millones de~      447.            717.            444.
## 10 1997 Millones de~      472.            775.            469.
## # ... with 22 more rows, and 2 more variables: masa_salarial_total_oede <dbl>,
## #   ms_cepal <dbl>
```

Activos

```
## # A tibble: 21 x 3
## # Groups:   empresa [1]
##   fecha      empresa tasa_depreciacion
##   <date>    <chr>          <dbl>
## 1 1998-12-31 YPF      0.115
## 2 1999-12-31 YPF      0.110
## 3 2000-12-31 YPF      0.102
## 4 2001-12-31 YPF      0.101
## 5 2002-12-31 YPF      0.114
## 6 2003-12-31 YPF      0.108
## 7 2004-12-31 YPF      0.120
## 8 2005-12-31 YPF      0.118
## 9 2006-12-31 YPF      0.160
## 10 2007-12-31 YPF      0.163
## # ... with 11 more rows
```

```
## # A tibble: 294 x 49
##   anio unidad      idsector sector presentaciones impgcia_c impgcia utilidad_c
##   <dbl> <chr>          <dbl> <chr>          <dbl>    <dbl>    <dbl>    <dbl>
```

```
## 1 2001 pesos cor~ 1 tot 124322 22743 4.28e 9 64479
## 2 2002 pesos cor~ 1 tot 124833 19775 6.44e 9 55446
## 3 2003 pesos cor~ 1 tot 126232 23142 1.28e10 64989
## 4 2004 pesos cor~ 1 tot 176766 33640 1.55e10 99548
## 5 2005 pesos cor~ 1 tot 182018 39525 1.94e10 105906
## 6 2006 pesos cor~ 1 tot 208065 49220 2.55e10 124645
## 7 2007 pesos cor~ 1 tot 187156 53003 3.07e10 119864
## 8 2008 pesos cor~ 1 tot 215040 62871 3.67e10 138394
## 9 2009 pesos cor~ 1 tot 231058 68919 4.13e10 144748
## 10 2010 pesos cor~ 1 tot 234158 76553 5.48e10 154847
## # ... with 284 more rows, and 41 more variables: utilidad <dbl>, perdida <dbl>,
## # perdida_c <dbl>, utilidadyperdida_c <dbl>, utilidad_neta <dbl>,
## # utilidadneta_ponderada <dbl>,
## # presentaciones - casos de utilidad y perdida <dbl>, activo_c <dbl>,
## # activo <dbl>, disponibilidades_c <dbl>, disponibilidades <dbl>,
## # creditos_c <dbl>, creditos <dbl>, bscambio_c <dbl>, bscambio <dbl>,
## # inventarios_c <dbl>, inventarios <dbl>, bsuso_c <dbl>, bsuso <dbl>, ...
```

```
## # A tibble: 84 x 6
##   anio sector          unidad          variable  valor fuente
##   <dbl> <chr>          <chr>          <chr>    <dbl> <chr>
## 1 2001 extraccion_petroleo_gas Millones de pesos 2018 KTA 457013. AFIP
## 2 2002 extraccion_petroleo_gas Millones de pesos 2018 KTA 778489. AFIP
## 3 2003 extraccion_petroleo_gas Millones de pesos 2018 KTA 723824. AFIP
## 4 2004 extraccion_petroleo_gas Millones de pesos 2018 KTA 905670. AFIP
## 5 2005 extraccion_petroleo_gas Millones de pesos 2018 KTA 701406. AFIP
## 6 2006 extraccion_petroleo_gas Millones de pesos 2018 KTA 979508. AFIP
## 7 2007 extraccion_petroleo_gas Millones de pesos 2018 KTA 919622. AFIP
## 8 2008 extraccion_petroleo_gas Millones de pesos 2018 KTA 872814. AFIP
## 9 2009 extraccion_petroleo_gas Millones de pesos 2018 KTA 781035. AFIP
## 10 2010 extraccion_petroleo_gas Millones de pesos 2018 KTA 668395. AFIP
## # ... with 74 more rows
```

```
## # A tibble: 45 x 14
##   anio sector unidad ventas costo_ventas utilidad_neta utilidad_operat~ pppe
##   <dbl> <chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 1967 Memor~ Millo~ NA NA 630. NA 1.67e5
## 2 1968 Memor~ Millo~ NA NA 11653. NA 1.72e5
## 3 1969 Memor~ Millo~ NA NA 9986. NA 1.76e5
## 4 1970 Memor~ Millo~ NA NA 663. NA 1.53e5
## 5 1971 Memor~ Millo~ NA NA 80096. NA 2.18e7
## 6 1972 Memor~ Millo~ NA NA -1748495. NA 1.09e7
## 7 1973 Memor~ Millo~ NA NA 1233134. NA 1.54e7
## 8 1974 Memor~ Millo~ NA 16527015. 2292298. 2451146. 1.21e7
## 9 1975 Memor~ Millo~ NA 18565972. -11297366. -11263523. 1.24e7
## 10 1976 Memor~ Millo~ NA 193095. -60637. -59751. 3.23e5
## # ... with 35 more rows, and 6 more variables: bienes_de_cambio <dbl>,
## # utilidad_neta_antes_impuestos <dbl>, activo <dbl>, patrimonio_netos <dbl>,
## # ipc_18 <dbl>, KTA <dbl>
```

Stock total de la rama

```

#metros perforados
metros_perforados_posterior_al_2009 <- read_csv("../data/secretaria_energia/sesco/metros-perforados.csv",
  col_types = cols(indice_tiempo = col_date(format = "%Y-%m")))

metros_perforados_anterior_al_2009 <- read_csv("../data/secretaria_energia/sesco/metros-perforados-anterior-al-2009.csv",
  rename(cantidad = Cantidad))

```

YPF

```
## Rows: 55608 Columns: 19
```

```

## -- Column specification -----
## Delimiter: ","
## chr  (12): idempresa, empresa, idareapermisococoncesion, areapermisococoncesion,...
## dbl  (5): anio, mes, idubicacion, idconcepto, Cantidad
## lgl  (1): observaciones
## date (1): fecha_data

```

```

##
## 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.

```

```

metros_perforados_empresa <- metros_perforados_posterior_al_2009 %>%
  bind_rows(metros_perforados_anterior_al_2009) %>%
  select(anio, mes, idempresa, empresa, idconcepto, concepto, cantidad, observaciones) %>%
  group_by(anio, idempresa, empresa) %>%
  summarise(metros_perforados = sum(cantidad, na.rm = T)) %>%
  ungroup()

```

```
## 'summarise()' has grouped output by 'anio', 'idempresa'. You can override using the '.groups' argument.
```

```

#listado de empresas con sus id's
listado_empresa <- distinct(metros_perforados_empresa %>%
  select(idempresa, empresa), idempresa, .keep_all = T)

# sec energia
# pozos cargados por empresas
listado_pozos <- read_csv("../data/secretaria_energia/cap_iv/listado-de-pozos-cargados-por-empresas-operando.csv",
  mutate(anio_terminacion_pozo = year(adjiv_fecha_fin_term)) %>%
  left_join(listado_empresa) %>%
  group_by(anio_terminacion_pozo, idempresa, empresa) %>%
  summarise(pozos = n()) %>%
  filter(!is.na(anio_terminacion_pozo))

```

```
## Rows: 79951 Columns: 51
```

```

## -- Column specification -----
## Delimiter: ","
## chr  (21): sigla, formprod, idempresa, idareapermisococoncesion, idareayacimie...

```



```
## dbl (22): idpozo, coordenadax, coordenaday, cota, profundidad, pet_inicial,...
## lgl (1): unique_sigla_formprod
## dtm (2): fechadeingreso, fecha_data
## date (5): adjiv_fecha_inicio, adjiv_fecha_fin, adjiv_fecha_inicio_term, adj...

##
## 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.

## Joining, by = "idempresa"
```

```
## 'summarise()' has grouped output by 'anio_terminacion_pozo', 'idempresa'. You can override using the
```

```
ppye_ypf <- ypf_seg %>%
  filter(sector == "upstream" ) %>%
  mutate(anio = year(fecha)) %>%
  left_join(ipc_promedio %>%
    select(anio, ipc_18), by = "anio") %>%
  mutate(variable = "activo_upstream_ypf",
    activos = activos/ipc_18,
    unidad = "Millones de pesos 2018") %>%
  select(anio, empresa, unidad , activo_upstream= activos) %>%
  left_join(stock_balances_empresas %>%
    ungroup() %>%
    filter(variable == "ppye" & empresa == "YPF") %>%
    select(anio, empresa, unidad, ppye_integrada =valor) , by = c("anio", "empresa", "unidad")) %>%
  mutate(activo_upstream = variacion_interanual(activo_upstream),
    ppye_integrada = variacion_interanual(ppye_integrada)) %>%
  full_join(listado_pozos %>%
    select(-empresa) %>%
    ungroup() %>%
    filter(idempresa == "YPF") %>%
    rename(anio = anio_terminacion_pozo, empresa = idempresa), by = c("anio", "empresa") ) %>%
  arrange(anio)
ppye_ypf
```

```
## # A tibble: 110 x 6
##   anio empresa unidad activo_upstream ppye_integrada pozos
##   <dbl> <chr>   <chr>         <dbl>         <dbl> <int>
## 1  1900 YPF     <NA>           NA            NA      1
## 2  1908 YPF     <NA>           NA            NA      1
## 3  1912 YPF     <NA>           NA            NA      1
## 4  1913 YPF     <NA>           NA            NA      1
## 5  1914 YPF     <NA>           NA            NA      1
## 6  1915 YPF     <NA>           NA            NA      1
## 7  1916 YPF     <NA>           NA            NA      4
## 8  1917 YPF     <NA>           NA            NA      2
## 9  1918 YPF     <NA>           NA            NA      2
## 10 1920 YPF     <NA>           NA            NA      3
## # ... with 100 more rows
```

```

ppye_ypf_tasas <- ppye_ypf %>%
  arrange(anio) %>%
  filter(anio %in% 1998:2018) %>%
  select(-c(anio, empresa, unidad)) %>%
  mutate_all(tasa_crecimiento)

#matriz de correlacion
# cor_pozos_ypf <- cor(ppye_ypf_tasas, use = "complete.obs")
# cor_pozos_ypf

# distintos correlogramas
#con ggcorrplot
# p.mat = cor_pmat(cor_pozos_ypf)
# ggcorrplot(cor_pozos_ypf, hc.order = TRUE,
#   type = "lower", p.mat = p.mat, lab = T)

# GGally::ggpairs(ppye_ypf_tasas, lower = list(continuous = "smooth"))
# plot(ppye_ypf_tasas)

#años bases
anio_base_ypf_segmento <- ppye_ypf$activo_upstream[ppye_ypf$anio == 2006]

anio_base_ypf_bolsar <- ppye_ypf$ppye_integrada[ppye_ypf$anio == 2006]

stock_y_pozos_ypf <- ppye_ypf %>%
  mutate(indice_pozos_06 = generar_indice(serie= pozos ,
                                          fecha = anio,
                                          fecha_base = 2006)) %>%
  mutate(ppye_ypf_estimado = anio_base_ypf_bolsar * indice_pozos_06,
         activo_ypf_estimado =anio_base_ypf_segmento * indice_pozos_06,
         unidad_ppye="Millones de pesos 2018")

# stock_y_pozos_ypf %>%
#   select(anio, unidad = unidad_ppye, ppye_ypf_estimado, activo_ypf_estimado) %>%
#   gather(key = estimacion,
#     value = valor,
#     3:4) %>%
#   filter(anio >1960) %>%
#   ggplot(aes(anio,valor , color = estimacion))+
#   geom_line()+
#   labs(y= "Millones de pesos de 2018", title = "Estimacion de Activo usptream y PPyE de YPF a partir

```

PPyE y activo upstream total rama a partir de YPF

```

pozos_yfp_sobre_total <- listado_pozos %>%
  rename(anio = anio_terminacion_pozo) %>%
  group_by(anio) %>%
  summarise(pozos_cargados_sec_energia = sum(pozos, na.rm = T)) %>%
  left_join(listado_pozos %>%
    rename(anio = anio_terminacion_pozo) %>%
    filter(idempresa == "YPF") %>%
    group_by(anio, idempresa) %>%

```

```
summarise(pozos_cargados_ypf = sum(pozos, na.rm = T))) %>%
mutate(ypf_sobre_total = pozos_cargados_ypf/ pozos_cargados_sec_energia)
```

'summarise()' has grouped output by 'anio'. You can override using the '.groups' argument.

Joining, by = "anio"

```
# pozos_ypf_sobre_total
```

```
estimacion_ppye_total_via_ypf <- stock_y_pozos_ypf %>%
  select(anio, unidad = unidad_ppye, ppye_ypf_estimado, activo_ypf_estimado) %>%
  gather(key = estimacion,
         value = valor,
         3:4) %>%
  left_join(pozos_ypf_sobre_total %>%
            select(anio, ypf_sobre_total)) %>%
  mutate(valor = valor/ypf_sobre_total,
         estimacion = case_when(estimacion == "ppye_ypf_estimado" ~ "ppye_estimado_total",
                                estimacion == "activo_ypf_estimado" ~ "activo_estimado_upstream" ))
```

Joining, by = "anio"

```
# estimacion_ppye_total_via_ypf %>%
#   filter(anio >1960) %>%
#   ggplot(aes(anio,valor , color = estimacion))+
#   geom_line()+
#   labs(y= "Millones de pesos de 2018",
#        title = "Estimacion de Activo usptream y PPyE total a partir de ampliación de pozos YPF")
```

Valor total de la producción

```
## # A tibble: 22 x 13
##   anio unidad  vbp_tot va_tot ci_tot vbp_extr va_extr ci_extr ms_tot ms_extr
##   <dbl> <chr>    <dbl> <dbl> <dbl>    <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 1993 Millones~    NA    NA    NA      NA     NA    NA    NA    NA
## 2 1994 Millones~    NA    NA    NA      NA     NA    NA    NA    NA
## 3 1995 Millones~    NA    NA    NA      NA     NA    NA    NA    NA
## 4 1996 Millones~    NA    NA    NA      NA     NA    NA    NA    NA
## 5 1997 Millones~    NA    NA    NA      NA     NA    NA    NA    NA
## 6 1998 Millones~    NA    NA    NA      NA     NA    NA    NA    NA
## 7 1999 Millones~    NA    NA    NA      NA     NA    NA    NA    NA
## 8 2000 Millones~    NA    NA    NA      NA     NA    NA    NA    NA
## 9 2001 Millones~    NA    NA    NA      NA     NA    NA    NA    NA
## 10 2002 Millones~    NA    NA    NA      NA     NA    NA    NA    NA
## # ... with 12 more rows, and 3 more variables: ebe_tot <dbl>, ebe_extr <dbl>,
## #   fuente <chr>

##   anio  coef_ci
## 1 1960      NA
## 2 1961      NA
```

##	3	1962	NA
##	4	1963	NA
##	5	1964	NA
##	6	1965	NA
##	7	1966	NA
##	8	1967	NA
##	9	1968	NA
##	10	1969	NA
##	11	1970	NA
##	12	1971	NA
##	13	1972	NA
##	14	1973	NA
##	15	1974	NA
##	16	1975	NA
##	17	1976	NA
##	18	1977	NA
##	19	1978	NA
##	20	1979	NA
##	21	1980	NA
##	22	1981	NA
##	23	1982	NA
##	24	1983	NA
##	25	1984	NA
##	26	1985	NA
##	27	1986	NA
##	28	1987	NA
##	29	1988	NA
##	30	1989	NA
##	31	1990	NA
##	32	1991	NA
##	33	1992	NA
##	34	1993	0.6205117
##	35	1994	0.6173273
##	36	1995	0.6161035
##	37	1996	0.6090919
##	38	1997	0.5987307
##	39	1998	0.5352324
##	40	1999	0.6126955
##	41	2000	0.6482184
##	42	2001	0.5950814
##	43	2002	0.3814898
##	44	2003	0.4162878
##	45	2004	0.3846116
##	46	2005	0.3744084
##	47	2006	0.3837072
##	48	2007	0.4411120
##	49	2008	0.4602018
##	50	2009	0.4180983
##	51	2010	0.4186047
##	52	2011	0.4606222
##	53	2012	0.4727875
##	54	2013	0.4941421
##	55	2014	0.3998503
##	56	2015	0.3959354

```
## 57 2016 0.3975447
## 58 2017 0.3890979
## 59 2018 0.3962114
## 60 2019 0.3981889
## 61 2020 0.4022803
```

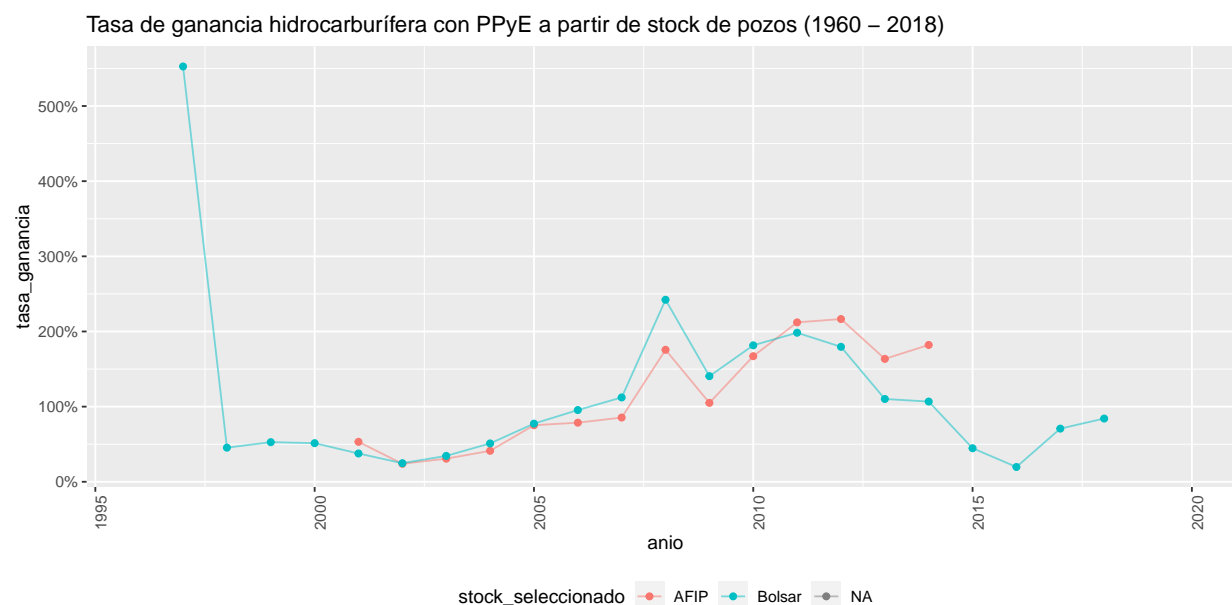
Estimaciones propias

```
## # A tibble: 110 x 32
## # Groups:   anio [110]
##   anio unidad_cant_crudo prod_crudo expo_crudo existencias_crudo
##   <dbl> <chr>           <dbl>      <dbl>          <dbl>
## 1 1911 barriles         12580.        0            0
## 2 1912 barriles         44029.        0            0
## 3 1913 barriles        132086.        0            0
## 4 1914 barriles        276751.        0            0
## 5 1915 barriles        515764.        0            0
## 6 1916 barriles        867992.        0            0
## 7 1917 barriles       1207642.        0            0
## 8 1918 barriles       1352307        0            0
## 9 1919 barriles       1327148.        0            0
## 10 1920 barriles       1647928.        0            0
## # ... with 100 more rows, and 27 more variables: unidad_precio_crudo <chr>,
## # precio_crudo_mdoint <dbl>, precio_me_crudo <dbl>, unidad_cant_gas <chr>,
## # prod_gas <dbl>, expo_gas <dbl>, unidad_precio_gas <chr>,
## # precio_externo_gas <dbl>, precio_exportacion_gas_ar <dbl>,
## # precio_gas_mdoint <dbl>, tcp <dbl>, tcc <dbl>, coef_ci <dbl>,
## # crudo_mdo_interno <dbl>, gas_mdo_interno <dbl>, unidad <chr>,
## # vbp_tot <dbl>, vbp_extr <dbl>, ci_tot <dbl>, ci_extr <dbl>, ...
```

```
## # A tibble: 110 x 17
##   anio unidad      fuente va_tot vbp_tot ms_tot ms_extr vbp_extr va_extr ci_tot
##   <dbl> <chr>      <chr>   <dbl>  <dbl>  <dbl>  <dbl>    <dbl>  <dbl>  <dbl>
## 1 1911 Millones ~ Empal~    NA    NA    NA    NA    NA    NA    NA
## 2 1912 Millones ~ Empal~    NA    NA    NA    NA    NA    NA    NA
## 3 1913 Millones ~ Empal~    NA    NA    NA    NA    NA    NA    NA
## 4 1914 Millones ~ Empal~    NA    NA    NA    NA    NA    NA    NA
## 5 1915 Millones ~ Empal~    NA    NA    NA    NA    NA    NA    NA
## 6 1916 Millones ~ Empal~    NA    NA    NA    NA    NA    NA    NA
## 7 1917 Millones ~ Empal~    NA    NA    NA    NA    NA    NA    NA
## 8 1918 Millones ~ Empal~    NA    NA    NA    NA    NA    NA    NA
## 9 1919 Millones ~ Empal~    NA    NA    NA    NA    NA    NA    NA
## 10 1920 Millones ~ Empal~    NA    NA    NA    NA    NA    NA    NA
## # ... with 100 more rows, and 7 more variables: ci_extr <dbl>, ebe_tot <dbl>,
## # ebe_extr <dbl>, coef_ci <dbl>, vbp_criterio_ccnn_04 <dbl>,
## # consumo_k_fijo <dbl>, pv <dbl>
```

Tasa de ganancia y renta apropiada por las empresas

Rentabilidad total de la rama



Renta apropiada por las empresas de la rama

```
## Joining, by = "anio"
```

Renta por el diferencial de precios entre el mercado interno y las referencias internacionales

```
## # A tibble: 108 x 13
##   anio unidad_cantidad prod_gas expo_gas prod_mdo_interno unidad_precio
##   <dbl> <chr>          <dbl>    <dbl>          <dbl> <chr>
## 1 1913 MMBTU           35300      0          35300 USD
## 2 1914 MMBTU          105900      0          105900 USD
## 3 1915 MMBTU          247100      0          247100 USD
## 4 1916 MMBTU          317700      0          317700 USD
## 5 1917 MMBTU          529500      0          529500 USD
## 6 1918 MMBTU          635400      0          635400 USD
## 7 1919 MMBTU          635400      0          635400 USD
## 8 1920 MMBTU          776600      0          776600 USD
## 9 1921 MMBTU          988400      0          988400 USD
## 10 1922 MMBTU         1306100      0         1306100 USD
## # ... with 98 more rows, and 7 more variables: precio_interno_gas <dbl>,
## #   precio_externo_gas <dbl>, tcc <dbl>, tcp <dbl>, unidad_renta <chr>,
## #   renta_dif_precios_gas <dbl>, renta_abaratamiento_sobrevaluacion_gas <dbl>
```

Renta apropiada por sobrevaluación cambiaria

```
## Joining, by = "anio"
```

```
## # A tibble: 110 x 13
## # Groups:   anio [110]
##   anio unidad_cantidad expo_crudo unidad_expo expo_crudo_usd unidad_precio
##   <dbl> <chr>          <dbl> <chr>          <dbl> <chr>
## 1 1911 barriles          0 <NA>          NA USD
## 2 1912 barriles          0 <NA>          NA USD
## 3 1913 barriles          0 <NA>          NA USD
## 4 1914 barriles          0 <NA>          NA USD
## 5 1915 barriles          0 <NA>          NA USD
## 6 1916 barriles          0 <NA>          NA USD
## 7 1917 barriles          0 <NA>          NA USD
## 8 1918 barriles          0 <NA>          NA USD
## 9 1919 barriles          0 <NA>          NA USD
## 10 1920 barriles          0 <NA>          NA USD
## # ... with 100 more rows, and 7 more variables: precio_externo_crudo <dbl>,
## #   tcc <dbl>, tcp <dbl>, unidad_renta <chr>, renta_sobrevaluacion_crudo <dbl>,
## #   renta_sobrevaluacion_crudo_valor <dbl>, dif <dbl>
```

Renta apropiada por el Estado mediante impuestos específicos

```
## Joining, by = c("anio", "unidad")
```

Renta Hidrocarburífera Total. En precios constantes, sobre Plus- valía Total y PBI

Método directo (a partir de descuentos sobre el VBP)

```
## Joining, by = "anio"
## Joining, by = "anio"
```

```
## # A tibble: 21 x 10
##   anio ipc_18 unidad      ppye tg_hidrocarburos tg_normal      pv subsidios
##   <dbl> <dbl> <chr>          <dbl>          <dbl> <dbl> <dbl>
## 1 1998 0.0397 Millones de ~ 8933.          0.142      0.136 6.85e4      0
## 2 1999 0.0392 Millones de ~ 8772.          0.139      0.108 8.31e4      0
## 3 2000 0.0389 Millones de ~ 17003.         0.0536      0.117 1.58e5      0
## 4 2001 0.0384 Millones de ~ 18999.         0.0468      0.114 1.14e5      0
## 5 2002 0.0484 Millones de ~ 30450.         0.318      0.189 6.54e4      0
## 6 2003 0.0549 Millones de ~ 28767.         0.361      0.196 1.03e5      0
## 7 2004 0.0573 Millones de ~ 30309.         0.454      0.186 1.90e5      0
## 8 2005 0.0628 Millones de ~ 35366.         0.446      0.182 3.47e5      0
## 9 2006 0.0697 Millones de ~ 39228.         0.568      0.180 4.45e5      0
## 10 2007 0.0788 Millones de ~ 40970.         0.486      0.178 4.95e5      0
## # ... with 11 more rows, and 2 more variables: gcia_normal_hidrocarburos <dbl>,
## #   renta_total <dbl>
```

Método indirecto (suma de mecanismos)

```
## Joining, by = "anio"
```

```

## Joining, by = "anio"
## Joining, by = "anio"
## Joining, by = "anio"

## 'mutate_all()' ignored the following grouping variables:
## Column 'anio'
## Use 'mutate_at(df, vars(-group_cols()), myoperation)' to silence the message.

## Joining, by = "anio"
## Joining, by = "anio"
## Joining, by = "anio"
## Joining, by = "anio"

```

Costos

```

#estimacion costos propia
costos <- produccion_total %>%
  select(anio, unidad_produccion=unidad,
         produccion_crudo =prod_crudo, produccion_gas=prod_gas,
         produccion_total=produccion_total_bep) %>%
  left_join(stock_estimado %>%
    left_join(ipc_promedio %>% select(anio, ipc_18) , by = "anio") %>%
    filter(fuente_ppye == "Bolsar") %>%
    mutate(consumo_k_fijo = valor * ipc_18 * consumo_k_fijo_ypf) %>%
    select(anio, consumo_k_fijo), by = "anio") %>%
  left_join(valor_total_produccion %>%
    # left_join(ipc_promedio %>% select(anio, ipc_18)) %>%
    left_join(tcp_anual_b %>% select(-sobrevaluacion) , by = "anio") %>%
    filter(variable %in% c("ci_extr", "ms_extr"),
           fuente%in% c( "Empalme CCNN")) %>% "CCNN oficial", "Criterio CCNN"
    mutate(valor = (valor*106)*ipc_18/tcc,
           unidad="USD tcc") %>%
    spread(key=variable, value=valor) , by = "anio") %>%
  left_join(renta_directo %>%
    select(anio, #stock_seleccionado,
           ppye, tg_normal, renta_total) , by = "anio") %>%
  left_join(precios_referencia_y_expo_crudo %>%
    select(anio, precio_referencia_externo = precio_me_crudo),by = c("anio")) %>%
  filter(!is.na(consumo_k_fijo)) %>%
  mutate(ppye = (ppye*106)*ipc_18/tcc,
         renta_total = (renta_total*106)*ipc_18/tcc,
         unidad_costos= "USD/boe",
         costos_totales = (ci_extr +ms_extr + consumo_k_fijo),
         costos_totales_sum_gcia_normal = costos_totales + (ppye * tg_normal),
         precio_costo = costos_totales/produccion_total,
         precio_costo_mmbtu = conversor_bepMMBTU_p(precio_costo),
         precio_produccion =costos_totales_sum_gcia_normal/produccion_total,
         precio_venta_potencial = (produccion_total *precio_referencia_externo -
                                   costos_totales)/produccion_total ) %>%
  select(anio, unidad_produccion,produccion_total ,
         unidad_ccnn = unidad, fuente, costos_totales,
         unidad_costos, precio_referencia_externo, costos_totales_sum_gcia_normal, precio_costo,

```



```

precio_costo_mmbtu, precio_produccion, precio_venta_potencial)
# precio_venta_potencial = ((ppye * tg_normal)+renta_total)/produccion_total)
# (produccion x precio internacional - costos_totales) / barriles BOE = gcia + renta por barril
#expresar en usd tcc y tcp
# costos

```

Comparación con estimación de otros autores

```

#renta usd vs autores
#falta agregar renta_hidrocarburos_jk
renta_comparacion <- renta_indirecto %>%
  select(anio,tcc, tcp, ipc_18, retenciones, regalias_total, contains("renta"), subsidios) %>%
  # select( -(13:ncol(.))) %>%
  select( -c(renta_pv, renta_pbi, renta_usd_tcc, renta_usd_tcp)) %>%
  filter(anio > 1962) %>%
  mutate( renta_usd_tcc = retenciones*ipc_18/tcc,
    regalias_usd_tcc = regalias_total*ipc_18/tcc,
    renta_diferencial_precios_crudo = renta_diferencial_precios_crudo*ipc_18/tcc,
    renta_diferencial_precios_gas = renta_diferencial_precios_gas*ipc_18/tcc,
    renta_expo_sobrevaluada_crudo = renta_expo_sobrevaluada_crudo*ipc_18/tcc,
    renta_expo_sobrevaluada_gas = renta_expo_sobrevaluada_gas*ipc_18/tcc ,
    renta_sobrevaluacion = renta_expo_sobrevaluada_crudo+ renta_expo_sobrevaluada_gas,
    renta_diferencial_precios =
      renta_diferencial_precios_gas+ renta_diferencial_precios_crudo,
    renta_empresas = renta_empresas*ipc_18/tcc,
    renta_total = (renta_total*ipc_18) / tcc,
    unidad = "Millones de USD",
    autor = "Propia") %>%
  rename(renta_usd_tcc = renta_usd_tcc,
    regalias_usd_tcc = regalias_usd_tcc,
    # renta_empresas = renta_empresas,
    # renta_total_gas = renta_diferencial_precios_gas,
    # renta_total_crudo = renta_diferencial_precios_crudo,
  ) %>%
  select(anio, unidad, autor, everything(.),
    -c(ipc_18, tcc, tcp, renta_expo_sobrevaluada_crudo, renta_expo_sobrevaluada_gas,
      renta_diferencial_precios_gas, renta_diferencial_precios_crudo)) %>%
  gather(key = tipo_de_renta,
    value = valor,
    4:ncol(.)) %>%
  bind_rows(renta_autores) %>%
  left_join(tcp_anual_b) %>%
  left_join(ipc_us) %>%
  mutate(valor = ((valor*tcc)/tcp)/ipc_us_20)

```

```

## Joining, by = "anio"
## Joining, by = "anio"

```

Exportacion de resultados

```
t1 <- Sys.time()
delta <- as.numeric( t1 - t0, units = "secs") #calculo la diferencia de tiempos
print( delta)
```

```
## [1] 80.07773
```