Taller de Consultoria - TP1

Gonzalo Barrera Borla

8/25/2019

Setup

```
library(fitdistrplus) # ajuste exploratorio de distribuciones
library(tidyverse) # manipulación de datos en general
library(broom) # limpieza y estructuracion de resultados de regresiones
library(RobStatTM) # regresiones robustas
```

Problema 1

Se dan las duraciones (medidas en ciclos hasta la ruptura) de una muestra de rodamientos ("rulemanes"). Describir las características principales de la muestra (posición, dispersión, asimetría), y buscar una distribución adecuada.

Los funcionales de locación más habituales son la media μ y la mediana η, digamos. Reportamos sus estimadores puntuales muestrales, $\hat{\mu} = \bar{x} = n^{-1} \sum_i x_i$ y $\hat{\eta} = x^{(\frac{n}{2})}$ (donde $x^{(i)}$ denota el i-ésimo elemento de la muestra ordenada). Para la dispersión, es razonable usar el la raíz cuadrada del estimador puntual insesgado de la varianza, $s^2 = (n-1)^{-1} \sum_i (x_i - \bar{x})^2$. Para la asimetría γ, construimos el estimador b reemplazando en la definición de γ a cada momento por su estimador insesgado:

$$\gamma = E\left[\left(\frac{X - \mu}{\sigma}\right)^3\right] \tag{1}$$

$$b = \hat{\gamma} = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{x_i - \bar{x}}{s}\right)^3 \tag{2}$$

```
df1 <- read_csv("data/1-1.csv")
asimetria <- function(x) {
  s <- sd(x)
  x_ <- mean(x) # "x raya"
  b <- ((x - x_) / s)^3
  return(mean(b))
}
df1 %>%
  summarise(
    media = mean(duracion),
    mediana = median(duracion),
    disp = sd(duracion),
    asim = asimetria(duracion)
    ) %>%
  knitr::kable(digits = 3)
```

media	mediana	disp	asim
72.387	61.68	38.363	0.847

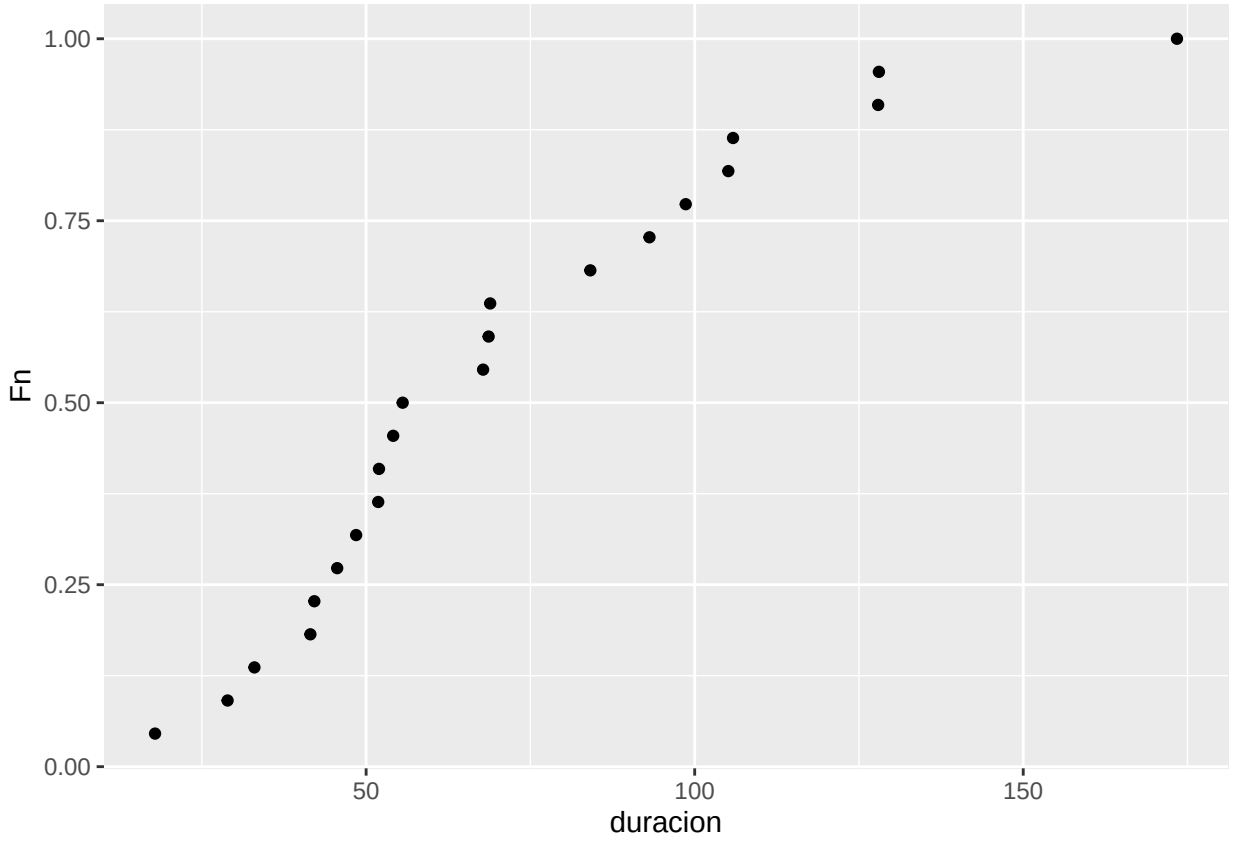
Comparamos nuestra funcion de asimetría con la implementación de un paquete bien conocido de R para comprobar su coherencia:

```
casi_iguales <- function(x, y, tol=1e-6) { abs(x-y) <= tol }
stopifnot(casi_iguales(
  e1071::skewness(df1$duracion),
  asimetria(df1$duracion)))
```

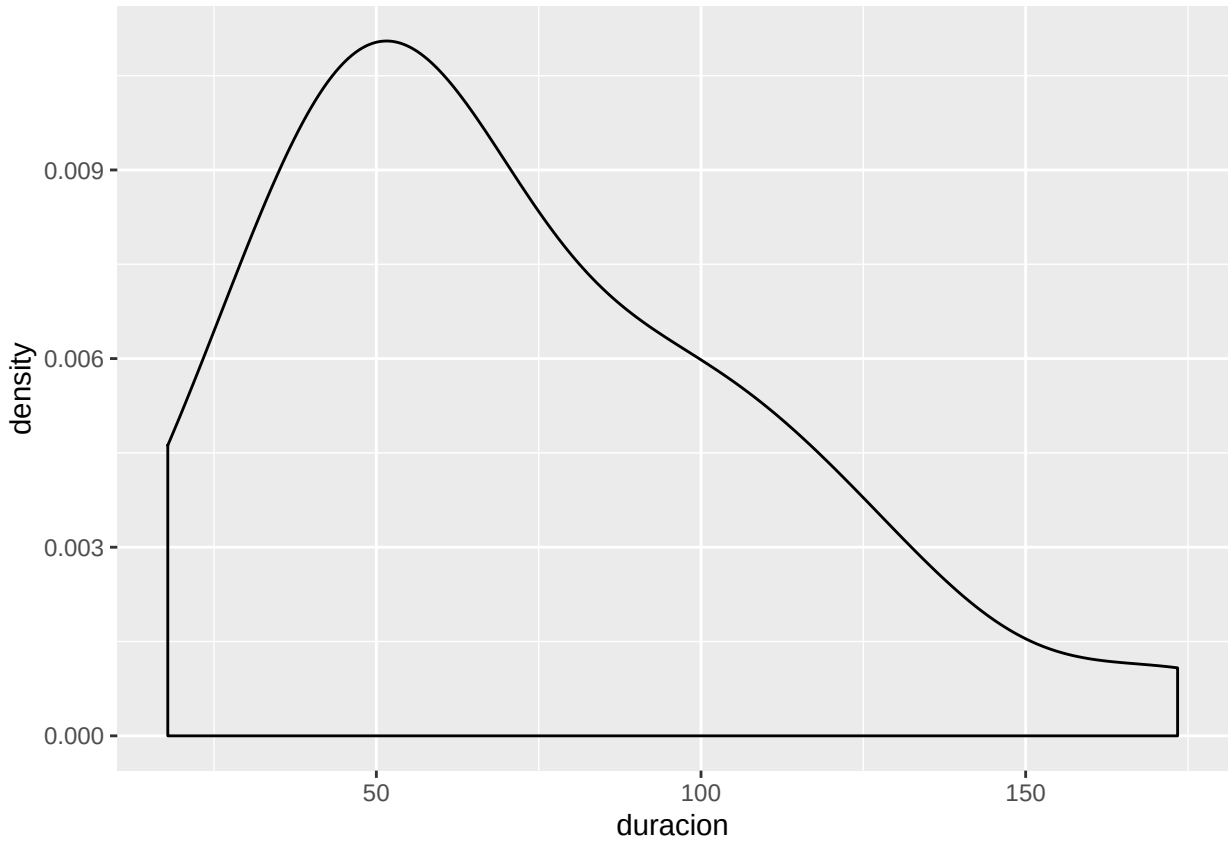
Investigamos gráficamente la posible distribución subyacente a los datos. Graficamos primero la distribución acumulada y densidad empírica, y luego usamos `fitdistrplus::descdist` para considerar posibles dustribuciones candidatas. Con tan pocos datos, es conveniente darse una idea de la dispersión de la asimetría y kurtosis estimadas a partir de la muestra, haciendo *bootstrap*. Vemos que tal vez una distribución beta o gamma sería adecuada.

Referencia: How to determine which distribution fits my data best?

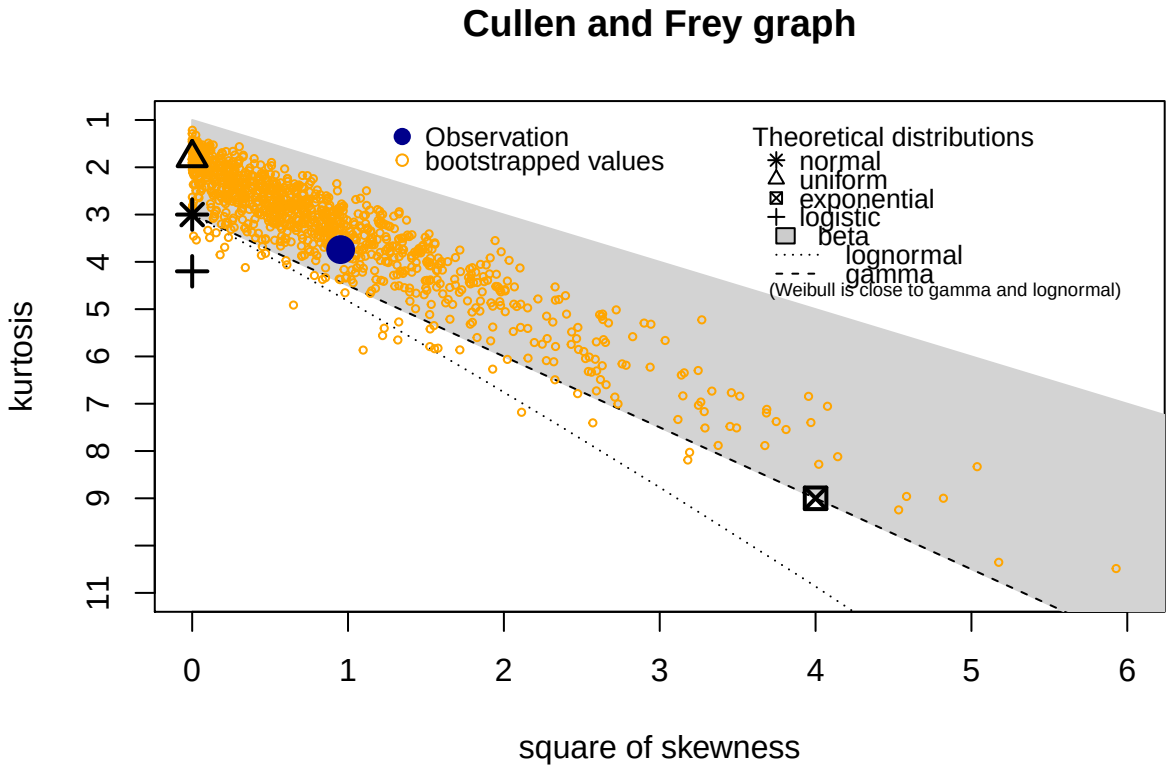
```
df1 %>%
  # Fn := distribucion empirica
  mutate(Fn = cume_dist(duracion)) %>%
  ggplot(aes(duracion, Fn)) +
  geom_point()
```




```
df1 %>%
  # Fn := distribucion empirica
  mutate(Fn = cume_dist(duracion)) %>%
  ggplot(aes(duracion)) +
  geom_density()
```




```
descdist(df1$duracion, boot = 1000)
```




```
## summary statistics
```

```
## ------
## min:  17.88   max:  173.4
## median:  61.68
## mean:  72.38727
## estimated sd:  38.36257
## estimated skewness:  0.9760312
## estimated kurtosis:  3.741007
```

Problema 2

Se dan: el punto de ebullición del agua (PE) (en grados Fahrenheit) y la presión atmosférica (PA) (en pulgadas de mercurio), medidos a distintas alturas en los Alpes. Plantear un modelo que describa cómo varía PE en función de PA. ¿Con cuánta precisión se puede estimar PE en función de PA?. Comentar cualquier característica de los datos.

Convertimos primero las unidades al sistema métrico, y las graficamos.

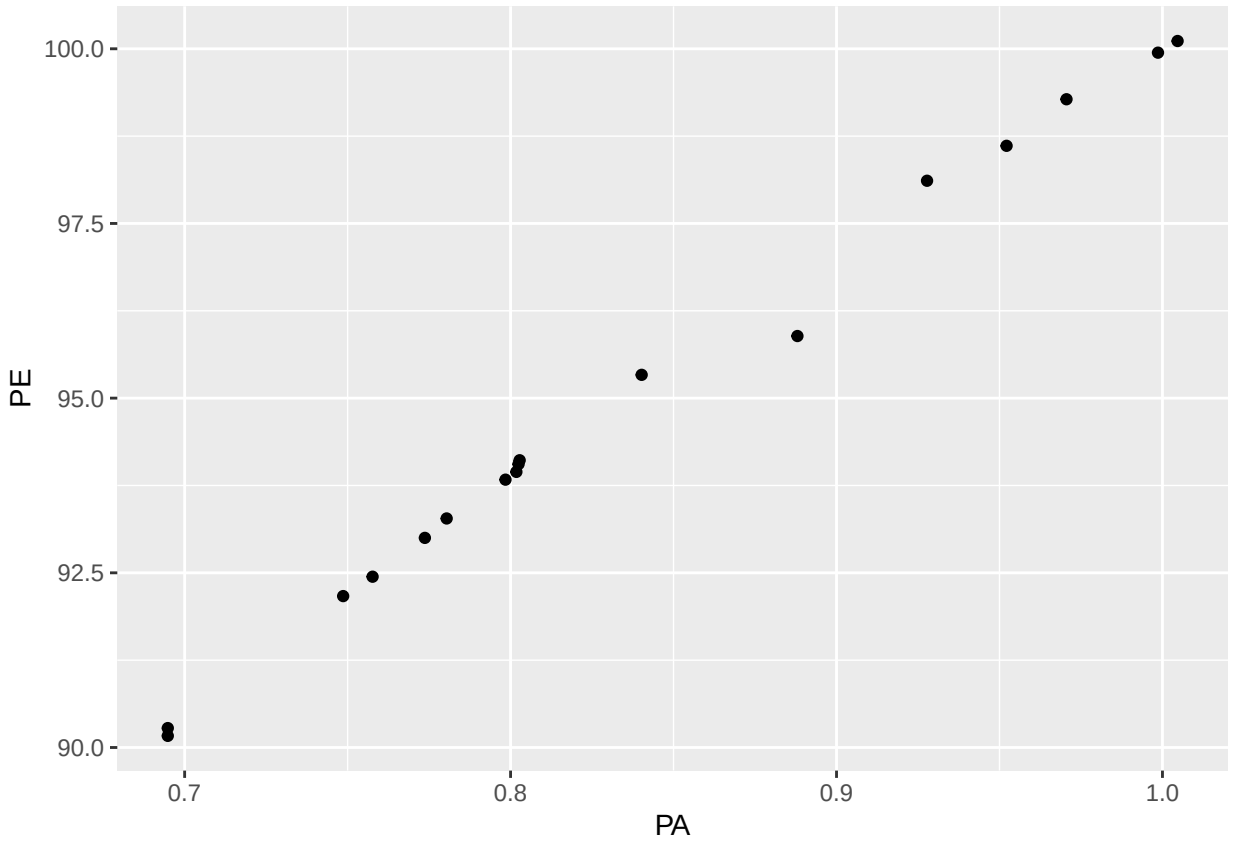
```
df2 <- read_csv("data/1-2.csv")
```

```
## Parsed with column specification:
## cols(
##   PE = col_double(),
##   PA = col_double()
## )
```

```
inHg_a_atm <- function(x) {x/29.921}
fahrenheit_a_celsius <- function(x) { (x - 32)*5/9 }

df2 <- df2 %>%
  mutate(
    id = seq_along(PA),
    PE = map_dbl(PE, fahrenheit_a_celsius),
    PA = map_dbl(PA, inHg_a_atm))

df2 %>%
  ggplot(aes(PA, PE)) +
  geom_point()
```



Se observa una relación casi lineal ($PE = b + m \times PA$), salvo por una tozuda observación cerca de (0.9, 95.0). Intentemos una sencilla regresión lineal en la PA, y grafiquemos los residuos en función de la misma.

```
lm2 <- lm(PE ~ PA, df2)
b <- round(lm2$coefficients[1], 2)
m <- round(lm2$coefficients[2], 2)

summary(lm2)
```

```
##
## Call:
## lm(formula = PE ~ PA, data = df2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.6816 -0.1232  0.0429  0.1094  0.2833
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  68.4980     0.5152  132.96   <2e-16 ***
## PA           31.6129     0.6110   51.74   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2467 on 15 degrees of freedom
## Multiple R-squared:  0.9944, Adjusted R-squared:  0.9941
## F-statistic:  2677 on 1 and 15 DF,  p-value: < 2.2e-16
```

Tanto la ordenada al origen como la pendiente parecen ser sin duda significativas, y nuestro modelo ajustado queda:

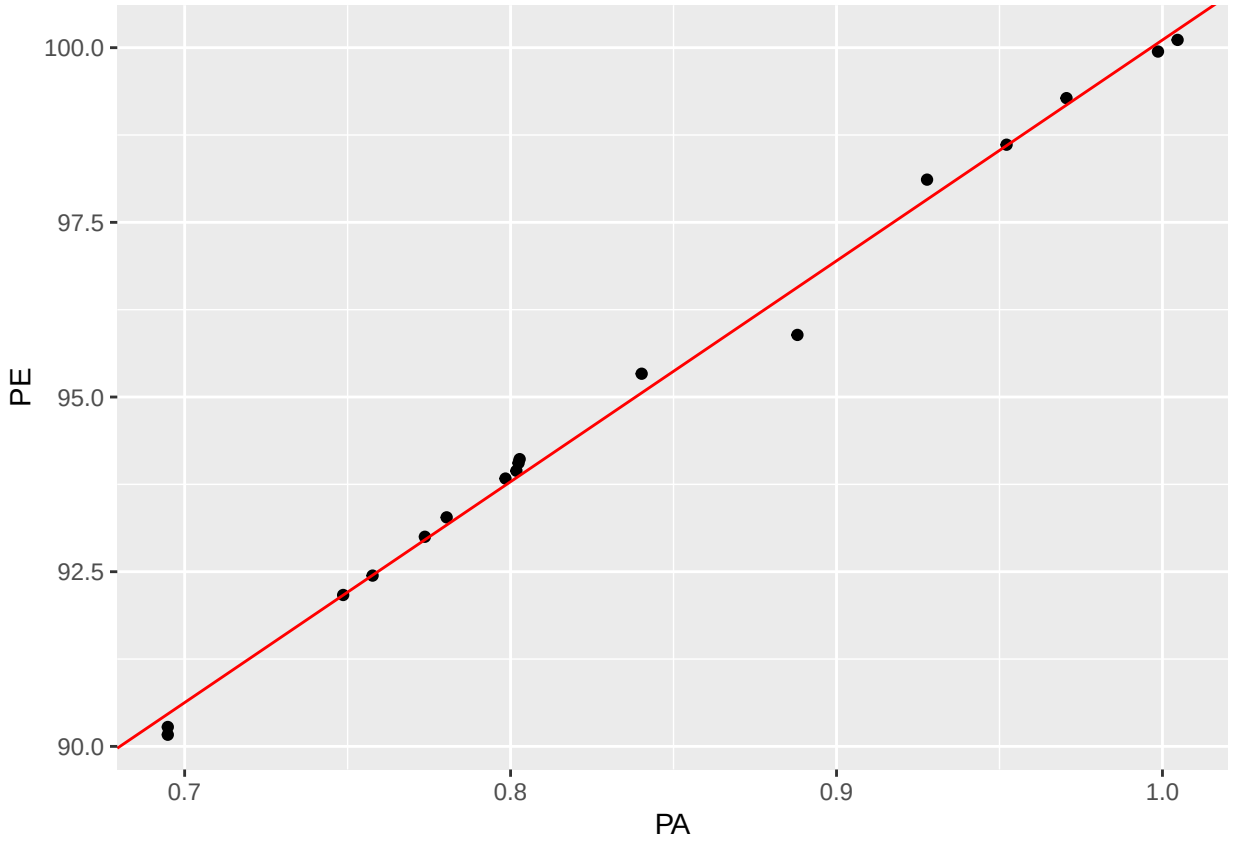
$$PE = 68.5°C + 31.61\frac{°C}{atm} \times PA$$

Vale aclarar que este modelo sólo tiene sentido sobre el soporte de los datos, es decir, cuando la presión atmosférica está en el rango $(0.69, 1)$. Si no, podríamos deducir incorrectamente que la temperatura de ebullición del agua en el vacío (0 atm.) es de 68.5 grados, cuando en realidad es de ~-68 grados celsius (referencia).

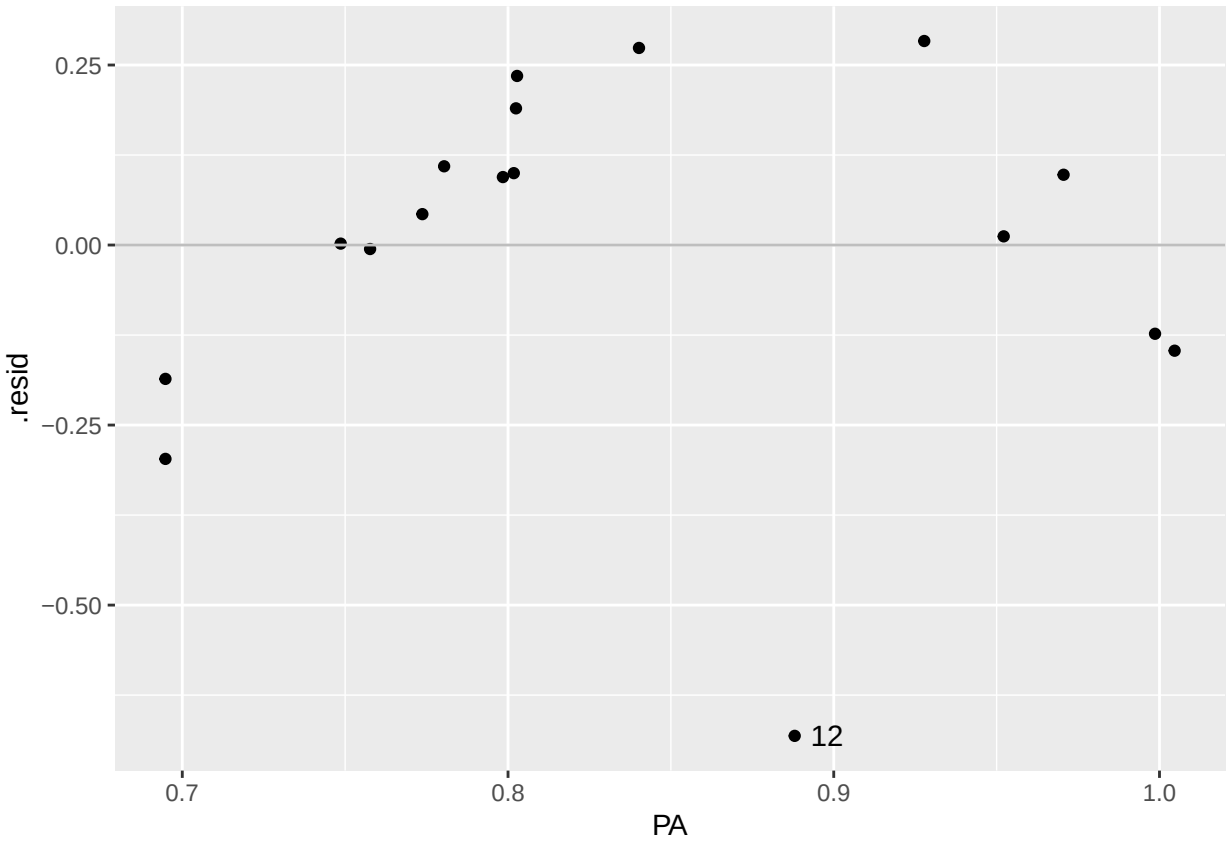
Observamos los "residuos" ($r_i = y_i - \hat{y}_i$) versus la presión atmosférica para ver con cuánta precisión se puede estimar el punto de ebullición, y examinarlos por si existe alguna estructura.

```
df2 <- augment(lm2, data = df2)

df2 %>%
  ggplot(aes(PA, PE)) +
  geom_point() +
  geom_abline(intercept = b, slope = m, color = 'red')
```



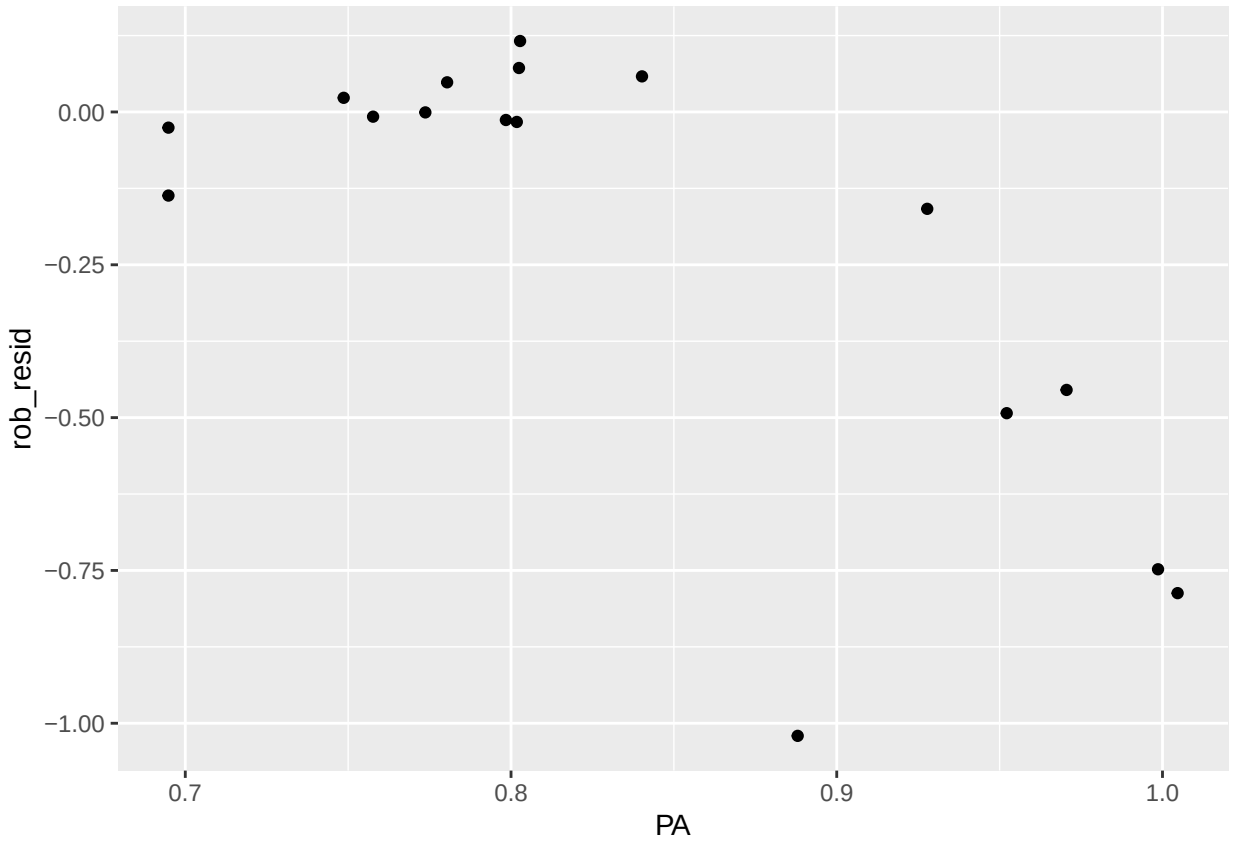

```
df2 %>%
  ggplot(aes(PA, .resid, label=id)) +
  geom_point() +
  geom_hline(yintercept = 0, color = 'gray') +
  geom_text(
    data = filter(df2, abs(.resid) > 0.5),
    nudge_x = 0.01)
```



Vemos que en general, la predicción del punto de ebullición está errada por menos de $0.25°C$, a excepción de la observación 12, donde el error es de -0.68 grados.

Se me ocurren dos caminos para mejorar este modelo. Uno, es realizar un ajuste robusto. Lo intentamos (con `RobStatTM::lmrobdetMM`), pero la situación no mejora demasiado. Una estrategia más sabia, sería plantear un modelo matemático con una descripción realista del fenómeno. Leyendo por la internet veo que la relación entre PA y PE es más bien exponencial, así que podríamos plantear un modelo tansformado acorde. Sin embargo, para el rango de PA considerado, pareciera má que razonable que la relación es lineal, y la santa observación #12 es más bien una mala medición de laboratorio.

```
cont <- lmrobdet.control(
  bb = 0.5, efficiency = 0.85, family = "bisquare")
lm2rob <- lmrobdetMM(PE ~ PA, data=df2, control=cont)
df2 %>%
  mutate(rob_resid = lm2rob$residuals) %>%
  ggplot(aes(PA, rob_resid)) +
  geom_point()
```



Problema 3

Se investiga el efecto de la presión aplicada durante la manufactura del papel, en el "factor de ruptura" (la fuerza necesaria para desgarrarlo). Bajo cada valor de la presión P, se manufacturó un lote de papel; de cada lote se eligieron 4 hojas, a cada una de las cuales se midió el factor de ruptura R. Se desea predecir R en función de P.

```
df3 <- read_csv("data/1-3.csv")
```

```
## Parsed with column specification:
## cols(
##   P = col_double(),
##   R = col_double()
## )
```

Aprovechando que tenemos mediciones repetidas para cada uno de los 5 valores de P, comparamos la raíz cuadrada del estimador global de la varianza $s_0 = 7.47$, con la de los estimadores de la varianza para cada valor de P:

```
df3 %>%
  group_by(P) %>%
  summarise(s = sd(R)) %>%
  knitr::kable(digits = 2)
```

P	s
35.0	3.30
49.5	7.97

P	s
70.0	7.72
99.0	4.80
140.0	2.63

Aún con pocos datos, se intuye que la varianza en R no es la misma para todo P. Pareciera haber *heterocedasticidad*, pero la relación entre P y la varianza de R no es lineal: s es máximo para presiones "medias" de fabricación.

Asumiendo que las mediciones de cada par (P, R) son independientes entre sí, la matriz de covarianzas será diagonal, y en vez de utilizar $\Sigma = \sigma^2 \mathrm{I}_n$, podemos considerar una matriz $\Sigma = diag(\sigma_1^2, ..., \sigma_n^2)$, y estimar las varianzas de cada observación, con el estimador insesgado de la varianza para cada nivel de presión antes calculado.

A continuación ajustamos ambos modelos, considerando (a) observaciones iid en general, y matriz de covarianza $\sigma^2 \mathrm{I}_n$, y (b) observaciones iid *en cada nivel de* P, con $\Sigma = diag(\sigma_1^2, ..., \sigma_n^2)$. Este último modelo equivale a realizar un ajuste de mínimos cuadrados pesados, con pesos $w_i = \sigma_i^{-2}$

```
df3 <-  df3 %>%
  group_by(P) %>%
  mutate(s = sd(R))

lm3a <- lm(R ~ P, df3)
lm3b <- lm(R ~ P, df3, weights = s^-2)

b <- lm3b$coefficients[1] %>% round(2)
m <- lm3b$coefficients[2] %>% round(2)

map_df(
  list(ordinario = lm3a, pesado = lm3b),
  glance, .id = 'modelo') %>%
  select(modelo, adj.r.squared, p.value) %>%
  knitr::kable()
```

modelo	adj.r.squared	p.value
ordinario	0.4692775	0.0005159
pesado	0.7563785	0.0000004

Aunque ambos modelos son buenos, el p-valor para la regresión global del segundo modelo es más de 3 órdenes de magnitud más pequeño. La evidencia parece justificar el uso de una regresión pesada. El modelo final quedará

$$R = 119.47 + -0.14 atm^{-1} \times P$$

NOTA: Graficar ambas rectas sobre los datos, ver que la diferencia es mínima.

Problema 4

La siguiente tabla da, para 12 huevos de gallina, la longitud L (o sea, el mayor diámetro), la mayor sección circular (el mayor diámetro perpendicular a L), ambas en pulgadas; y el volumen V. Interesa predecir V en función de L y M.

Preston (1973) (link) hace un tratamiento bastante exhaustivo de cómo calcular el volumen de un huevo, que a continuación resumimos.

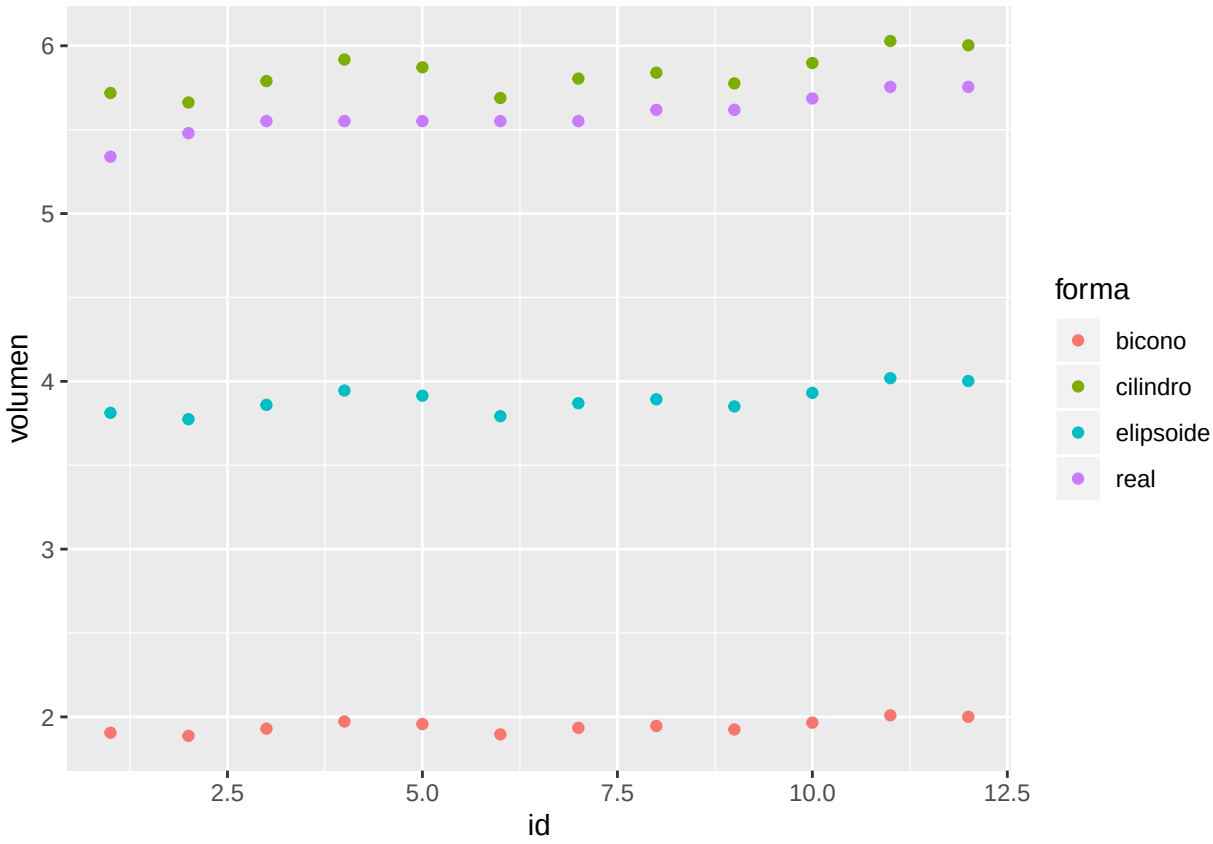
Supongamos que el mayor diámetro de un huevo es M, y su largo es $L = a + b$, donde a, b son las dos partes en que se divide el largo a la altura del máximo diámetro. El volumen de todo huevo está acotado superiormente por un ciindro perfecto de diámetro M, y por debajo por un bicono de máximo diámetro M y alturas a, b. Si el huevo fuese cilíndrico, su volumen será $\frac{\pi}{4}M^2L$, y si fuese bicónico, $\frac{\pi}{12}M^2L$. En un escenario más realista e intermedio, en que el huevo está formado por dos medios elipsoides, su volumen es $\frac{\pi}{6}M^2L$. Nótese que en ningún caso la asimetría (*id est*, cuán lejos de $L/2$ están a y b) hace diferencia alguna, pero sí es clave saber la forma dominante (bicono, elipsoide, cilindro). Los huevos de colibrí son más bien romos, casi cilíndricos, mientras que los de zampullín son casi bicónicos. En el siguiente gráfico, exhibimos los posibles volúmenes de cada huevo según la suposición de forma:

```
df4 <- read_csv("data/1-4.csv")
```

```
## Parsed with column specification:
## cols(
##   L = col_double(),
##   M = col_double(),
##   V = col_double()
## )
```

```
volumenes <- df4 %>%
  arrange(V) %>%
  mutate(
    real = V,
    id = seq_along(V),
    bicono = pi/12 * M^2 * L,
    elipsoide = pi/6 * M^2 * L,
    cilindro = pi/4 * M^2 * L) %>%
  select(id, real, bicono, elipsoide, cilindro) %>%
  gather(forma, volumen, -id)

volumenes %>%
  ggplot(aes(id, volumen, color = forma)) +
  geom_point()
```



El formato más razonable parece ser un cilindro, así que si el volumen del huevo de gallina está dado por Íla fórmula $V = kM^2L$, $k \approx \pi/4$. Sin embargo, una simple regresión lineal sobre M y L que incluya un término cuadrático sobre M ya tiene un error cuadrático medio mucho menor que nuestro modelo de huevo cilíndrico:

```
lm4a <- lm(V ~ poly(M, 2) + L, df4)
df4a <- augment(lm4a, data = df4)
df4a %>%
  mutate(cilindro = pi/4 * M^2 * L) %>%
  summarise(
    "ecm_lm" = mean(.resid^2),
    ecm_cil = mean((V - cilindro)^2)) %>%
  knitr::kable(digits = 4)
```

ecm_lm	ecm_cil
0.0043	0.0674

Una forma directa de mejorar el modelo, es usar una regresión lineal *sin ordenada*, sobre una covariable "sintética", $V = k \times (M^2 \cdot L)$ y estimar empíricamente k.

```
df4 <- df4 %>%
  mutate(M2L = M^2*L)
lm4b <- lm(V ~ M2L + 0, df4)
k <- lm4b$coefficients[1] %>% round(3)
ecm <- function(lm_call) { mean(lm_call$residuals^2) }
map_df(
  list(polinomico = lm4a, fisico = lm4b),
  glance, .id = 'modelo') %>%
```

```
  select(modelo, adj.r.squared, p.value) %>%
  knitr::kable(digits = 4)
```

modelo	adj.r.squared	p.value
polinomico	0.5164	0.0319
fisico	0.9998	0.0000

¡Y cómo mejora! Evidentemente, con un R^2_{adj} tan cercano a 1, algo debemos haber hecho bien. Es interesante notar que si comparásemos los dos modelos según su error cuadrático medio, ,el "polinómico" ingenuo da 0.0043 y el "físico" basado en una teoría real sobre la forma de los huevos de aves, da 0.0051. Las predicciones del modelo polinómico tienen menor error cuadrático medio, pero el modelo físico es tanto más parsimonioso (ajusta 1 sólo parámetro en lugar de 4), que termina siendo ampliamente preferible. Así, concluimos que el mejor modelo para predecir V es $V = 0.752 \times M^2 \times L$.