Linear regression II

2024-09-05

```
stopifnot(
  require(tidyverse),
  require(patchwork),
  require(httr),
  require(glue),
  require(broom)
)
old_theme <- theme_set(theme_minimal())
```

- **M1 MIDS/MFA**
- **Université Paris Cité**
- Année 2024-2025
- Course Homepage
- Moodle

Objectives

Linear fit using ordinary least squares (OLS)

- Perform linear regression of SAL_ACTUEL with respect to SAL_EMBAUCHE. Store the result in an object denoted by `lm_1`
- Inspect the numerical summary of `lm_1`
- Use `Environment` panel (Rstudio), to explore the structure of `lm_1`. Try to understand the signification of each element.

```
datapath <- '../DATA'
fname <- 'Banque.csv'
fpath <- paste(datapath, fname, sep="/")
```

```
if (!file.exists(fpath)) {
  baseurl <- 'https://stephane-v-boucheron.fr/data'
  download.file(url=paste(baseurl, fname, sep="/"),
                destfile=fpath)
  print(glue::glue('File {fname} downloaded at {fpath}!'))
} else {
  print(glue::glue('File {fname} already exists at {fpath}!'))
}
```

```
File Banque.csv already exists at ../DATA/Banque.csv!
```

```
bank <- readr::read_table(fpath,
    col_types = cols(
        SEXE = col_factor(levels = c("0", "1")),
        CATEGORIE = col_integer(),
        NB_ETUDES = col_integer(),
        SATIS_EMPLOI = col_factor(levels = c("non", "oui")),
        SATIS_CHEF = col_factor(levels = c("non", "oui")),
        SATIS_SALAIRE = col_factor(levels = c("non", "oui")),
        SATIS_COLLEGUES = col_factor(levels = c("non", "oui")),
        SATIS_CE = col_factor(levels = c("non", "oui"))
  )
)
```

```
lm_1 <- lm(formula = SAL_ACTUEL ~ SAL_EMBAUCHE, data=bank)

lm2str_frm <- . %>%
  formula() %>%
  deparse()

frm_1 <- lm2str_frm(lm_1)

summary(lm_1)
```

```
Call:
lm(formula = SAL_ACTUEL ~ SAL_EMBAUCHE, data = bank)

Residuals:
   Min     1Q Median     3Q    Max
-35424  -4031  -1154   2584  49293

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.928e+03  8.887e+02    2.17   0.0305 *
SAL_EMBAUCHE 1.909e+00  4.741e-02   40.28   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 8115 on 472 degrees of freedom
Multiple R-squared:  0.7746,    Adjusted R-squared:  0.7741
F-statistic:  1622 on 1 and 472 DF,  p-value: < 2.2e-16
```

```
cor(lm_1$fitted.values, bank$SAL_ACTUEL)^2
```

```
[1] 0.7746068
```

```
var(lm_1$fitted.values)/var(bank$SAL_ACTUEL)
```

```
[1] 0.7746068
```

- Make the model summary a dataframe/tibble using `broom::tidy()`

```
lm_1 %>%
  tidy() %>%                                          (1)
  knitr::kable(digit=2, caption = frm_1)
```

(1) `tidy()` is a *generic* function that can be applied to very different classes of objects

Table 1: SAL_ACTUEL ~ SAL_EMBAUCHE

term	estimate	std.error	statistic	p.value
(Intercept)	1928.21	888.68	2.17	0.03
SAL_EMBAUCHE	1.91	0.05	40.28	0.00

- Make model diagnostic information a dataframe/tibble using `broom::glance()`

```r
lm_1 %>%
  glance() %>%                                    ①
  knitr::kable(digit=2, caption = frm_1)
```

① `glance` is also a generic function

Table 2: SAL_ACTUEL ~ SAL_EMBAUCHE

r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance	df.residual	nobs
0.77	0.77	8115.36	1622.12	0	1	-4938.29	9882.58	9895.07	31085446686	472	474

- Preparing for diagnostic plots using `broom::augment()`

```r
lm_1_aug <- lm_1 %>%                              ①
  augment(data=bank)                              ②

lm_1_aug %>%
  DT::datatable(extensions = "Responsive")
```

① `lm_1` is list with many named components
② The output of `augment` is a dataframe built from informations gathered in `lm_1` and in `bank`

```
PhantomJS not found. You can install it with webshot::install_phantomjs(). If it is instal
```

Show 10 entries   Search:

	SEXE	AGE	CATEGORIE	NB_ETUDES	EXPERIENCE	ANCIENNETE	SAL_EMBAUCHE	SAL_ACTUEL
▶ 1	1	64	1	8	275	70	10200	15750
▶ 2	1	55	1	8	43	74	10200	15900
▶ 3	1	56	1	8	0	92	9750	16200
▶ 4	1	60	1	12	0	82	10200	16200
▶ 5	1	62	1	12	180	68	10200	16200
▶ 6	1	61	1	12	163	66	10200	16350
▶ 7	1	62	1	12	288	84	10200	16500
▶ 8	1	63	1	8	412	88	9750	16650
▶ 9	1	59	1	8	76	76	10200	16800
▶ 10	1	61	1	12	124	97	9000	16950

Showing 1 to 10 of 474 entries   Previous 1 2 3 4 5 … 48 Next

The output of `augment` may be described as adding 6 columns to dataframe `bank`. The six columns are built using items from `lm_1`. Can you explain their meaning and why they are relevant to diagnosing?

```
lm_1_aug %>%
  select(starts_with(".")) %>%
  head() %>%
  knitr::kable(digits=2, caption = frm_1)
```

Table 3: SAL_ACTUEL ~ SAL_EMBAUCHE

.fitted	.resid	.hat	.sigma	.cooksd	.std.resid
21404.59	-5654.59	0	8119.77	0	-0.70
21404.59	-5504.59	0	8119.99	0	-0.68
20545.34	-4345.34	0	8121.49	0	-0.54
21404.59	-5204.59	0	8120.41	0	-0.64
21404.59	-5204.59	0	8120.41	0	-0.64
21404.59	-5054.59	0	8120.61	0	-0.62

Let base `R` produce diagnostic plots

```
plot(lm_1, which = 1:6)
```

We will reproduce (and discuss) four of the six diagnostic plots provided by the `plot` method from base `R` (1,2,3,5).

- Reproduce first diagnostic plot with `ggplot` using the aumented version of `lm_1` (`augment(lm_1)`).

```
p_1_lm_1 <- lm_1_aug %>%
  ggplot() +
  aes(x=.fitted, y=.resid)+
  geom_point(alpha=.5, size=.5) +
  geom_smooth(formula = y ~ x,
              method="loess",
              se=F,
              linetype="dotted",
              linewidth=.5,
              color="black") +
  xlab("Fitted values") +
  ylab("Residuals)") +
  ggtitle("Bank dataset",
          subtitle = frm_1) +
  labs(caption = "Residuals versus Fitted")
```

- Comment Diagnostic Plot 1.
- Compute the correlation coefficient between residuals and fitted values.
- Make your graphic pipeline a reusable function.

```
make_p_diag_1 <- function(lm.){
  augment(lm.) %>%
  ggplot() +
  aes(x=.fitted, y=.resid)+
  geom_point(alpha=.5, size=.5) +
```

```
  geom_smooth(method="loess",
              formula = y ~ x,
              se=F,
              linetype="dotted",
              size=.5,
              color="black") +
  xlab("Fitted values") +
  ylab("Residuals)") +
  labs(title = "Residuals versus Fitted")
}
```

- What are *standardized residuals* ?
- Build the third diagnostic plot (square root of absolute values of standardized residuals versus fitted values) using `ggplot`.
- Why should we look at the square root of standardized residuals?

```
p_3_lm_1 <- lm_1_aug %>%
  ggplot() +
  aes(x=.fitted, y=sqrt(abs(.std.resid))) +
  geom_smooth(formula = y ~ x,
              se=F,
              method="loess",
              linetype="dotted",
              size=.5,
              color="black") +
  xlab("Fitted values") +
  ylab("sqrt(standardized residuals)") +
  geom_point(size=.5, alpha=.5) +
  ggtitle("Bank dataset",
          subtitle = frm_1) +
  labs(caption = "Scale location")
```

```
Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
i Please use `linewidth` instead.
```

Make your graphic pipeline a reusable function.

```
make_p_diag_3 <-  function(lm.){
  augment(lm.) %>%
  ggplot() +
  aes(x=.fitted, y=sqrt(abs(.std.resid))) +
  geom_smooth(formula = y ~ x,
              method="loess",
              se=F,
              linetype="dotted",
              size=.5,
              color="black") +
  xlab("Fitted values") +
  ylab("sqrt(standardized residuals)") +
  geom_point(size=.5, alpha=.5) +
  labs(title = "Scale location")
}
```

- What is leverage ?
- Build the fifth diagnostic plot (standardized residuals versus leverage) using `ggplot`.

```
p_5_lm_1 <- lm_1_aug %>%
  ggplot() +
  aes(x=.hat, y=((.std.resid))) +
  geom_point(size=.5, alpha=.5) +
  xlab("Leverage") +
  ylab("Standardized residuals") +
  ggtitle("Bank dataset",
          subtitle = frm_1)

# plot(lm.1, which = 5)
```

```
make_p_diag_5 <-  function(lm.){
  augment(lm.) %>%
  ggplot() +
  aes(x=.hat, y=((.std.resid))) +
  geom_point(size=.5, alpha=.5) +
  xlab("Leverage") +
  ylab("Standardized residuals") +
  labs(title = "Standardized residulas versus Leverages")
}
```

In the second diagnostic plot (the residuals qqplot), we build a quantile-quantile plot by plotting function $F_n^{\leftarrow} \circ \Phi$ where Φ is the ECDF of the standard Gaussian distribution while $F_n^{\leftarrow}$.

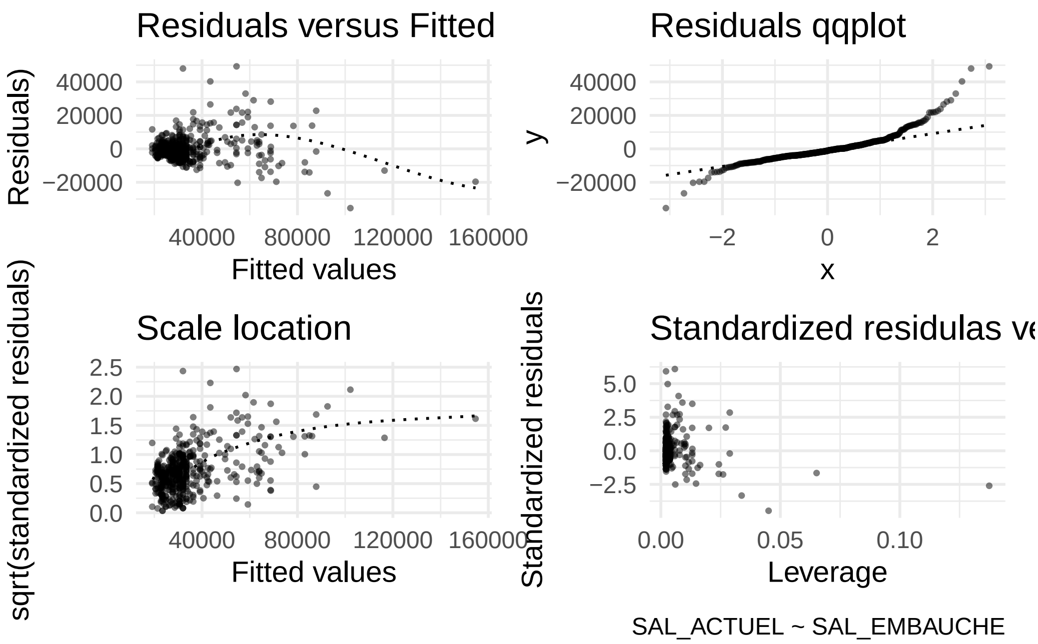
Build the second diagnostic plot using `ggplot`

```
p_2_lm_1 <- lm_1_aug %>%
  ggplot() +
  aes(sample=.resid) +
  geom_qq(size=.5, alpha=.5) +
  stat_qq_line(linetype="dotted",
               size=.5,
               color="black") +
  ggtitle("Bank dataset",
          subtitle = frm_1) +
  labs(caption="Residuals qqplot") +
  xlab("Theoretical quantiles") +
  ylab("Empirical quantiles of residuals")

# plot(lm_1, which = 2)
```

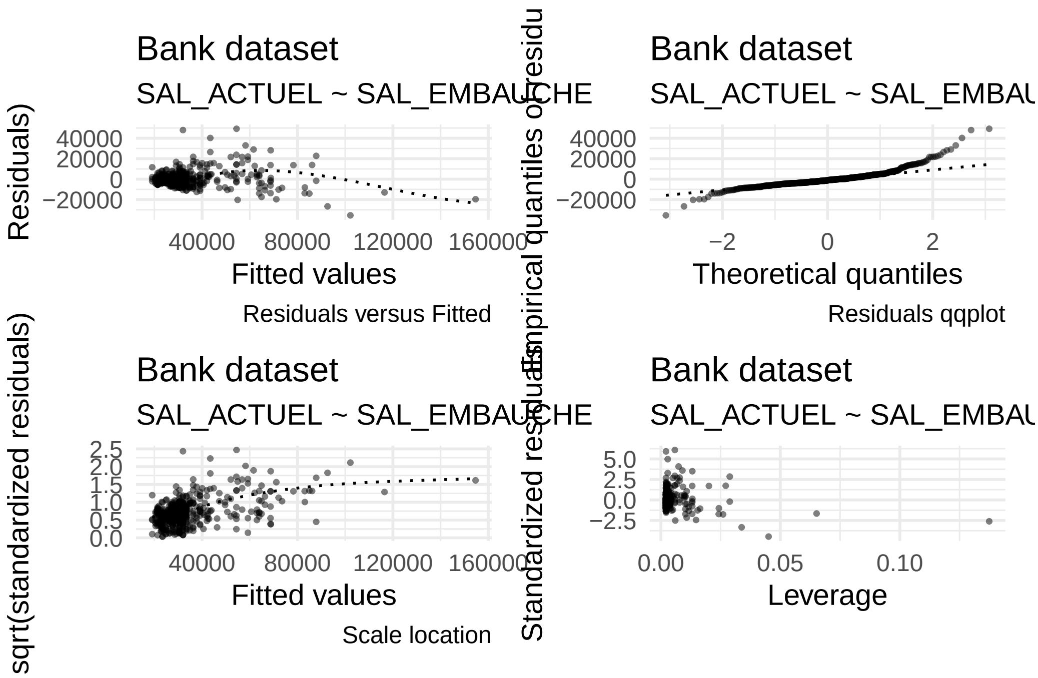
```
make_p_diag_2 <-  function(lm.){
  augment(lm.) %>%
  ggplot() +
  aes(sample=.resid) +
  geom_qq(size=.5, alpha=.5) +
  stat_qq_line(linetype="dotted",
               size=.5,
               color="black") +
  labs(title="Residuals qqplot")
}
```

Use package `patchwork::...` to collect your four diagnostic plots

```
lyt <- patchwork::plot_layout(ncol=2, nrow=2)

(make_p_diag_1(lm_1) +
make_p_diag_2(lm_1) +
make_p_diag_3(lm_1) +
make_p_diag_5(lm_1) ) +
  patchwork::plot_annotation(caption='SAL_ACTUEL ~ SAL_EMBAUCHE') # DRY this ?
```



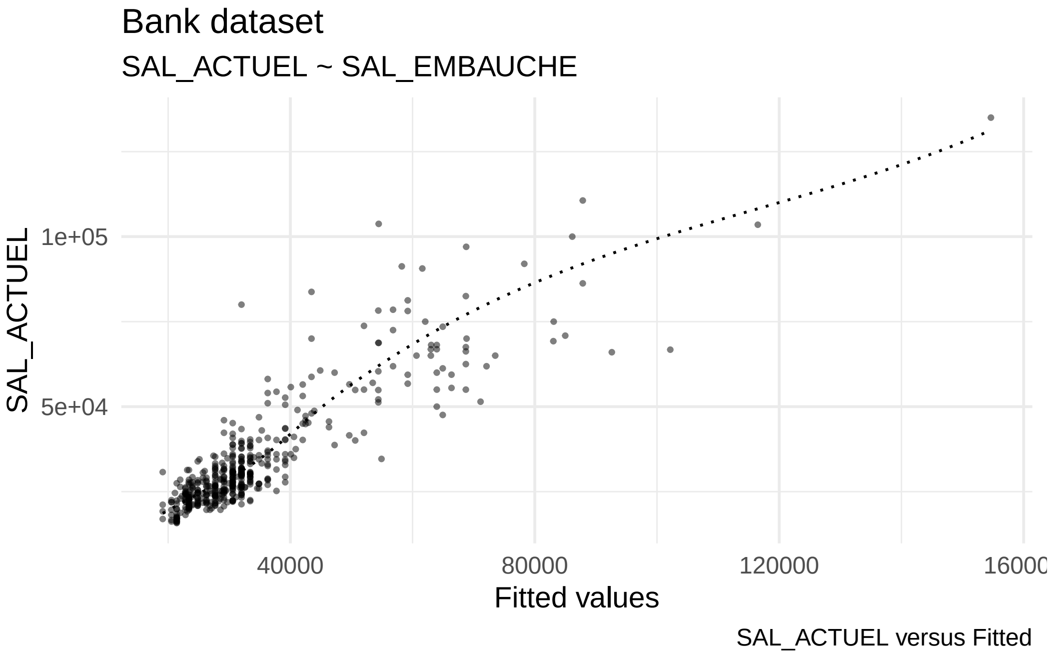

```
p_1_lm_1 + p_2_lm_1 + p_3_lm_1 + p_5_lm_1
```



Plot actual values against fitted values for `SAL_ACTUEL`

```
p_1_bis_lm_1 <- lm_1_aug %>%
  ggplot() +
```

```
  aes(x=.fitted, y=SAL_ACTUEL)+
  geom_point(alpha=.5, size=.5) +
  geom_smooth(formula = y ~ x,
              method="loess",
              se=F,
              linetype="dotted",
              size=.5,
              color="black") +
  xlab("Fitted values") +
  ylab("SAL_ACTUEL") +
  ggtitle("Bank dataset",
          subtitle = frm_1) +
  labs(caption = "SAL_ACTUEL versus Fitted")

p_1_bis_lm_1
```



Play it again with `AGE` and `SAL_ACTUEL`

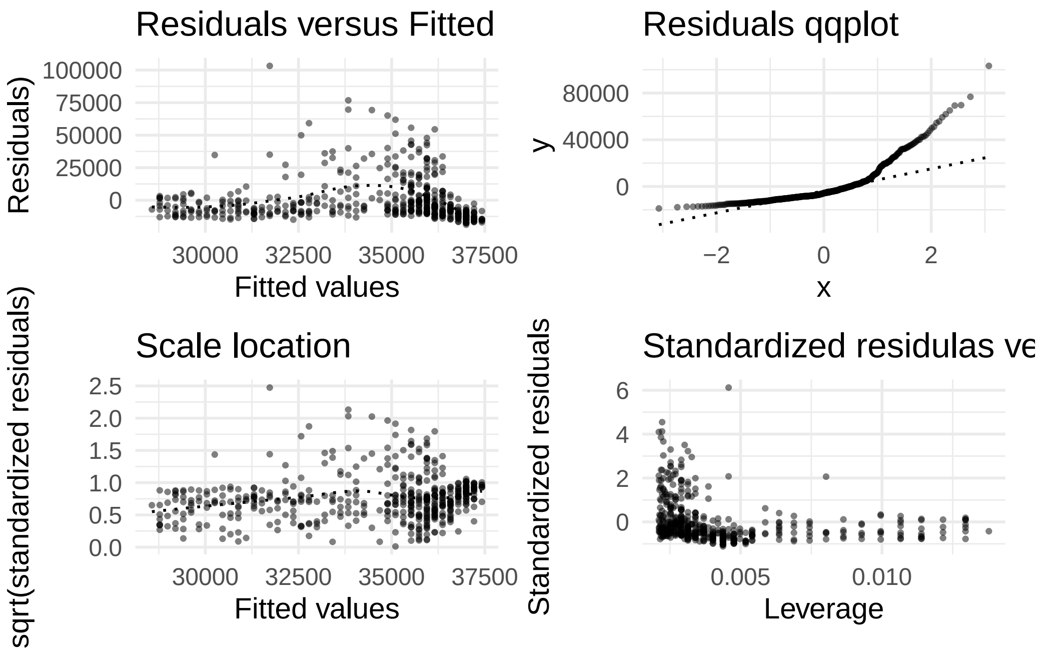
Redo the above described steps and call the model `lm_2`.

```
lm_2 <- lm(SAL_ACTUEL ~ AGE, data=bank)

lm_2 %>%
  tidy()
```

```
# A tibble: 2 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)   42272.     2571.     16.4  2.30e-48
2 AGE            -211.       65.8    -3.20 1.45e- 3
```

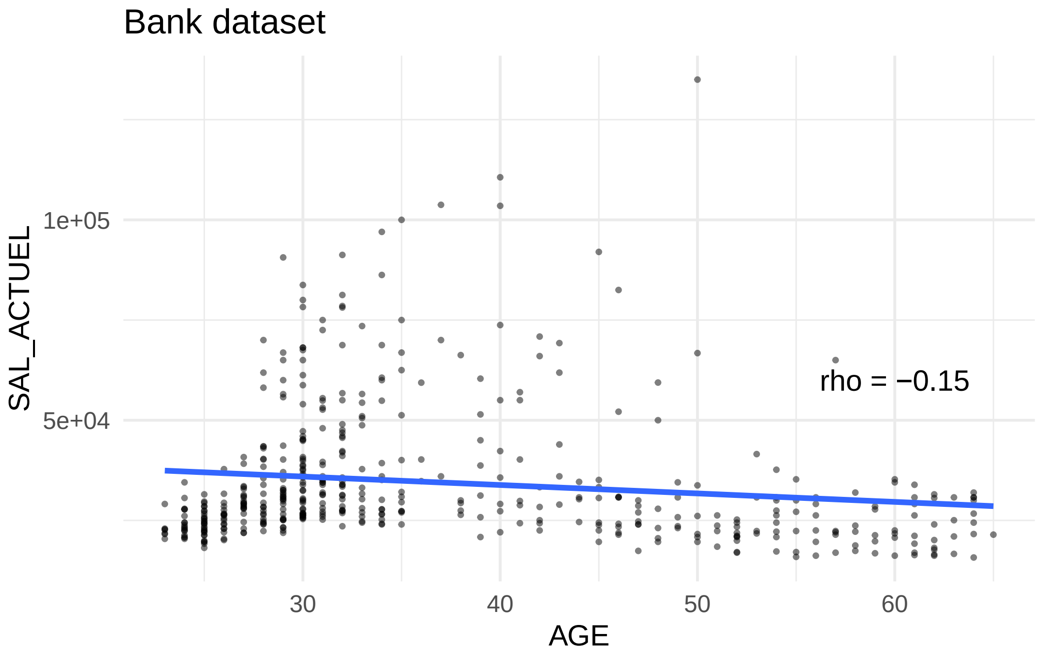
```
lyt <- patchwork::plot_layout(ncol=2, nrow=2)

make_p_diag_1(lm_2) +
  make_p_diag_2(lm_2) +
```

```
make_p_diag_3(lm_2) +
make_p_diag_5(lm_2)
```

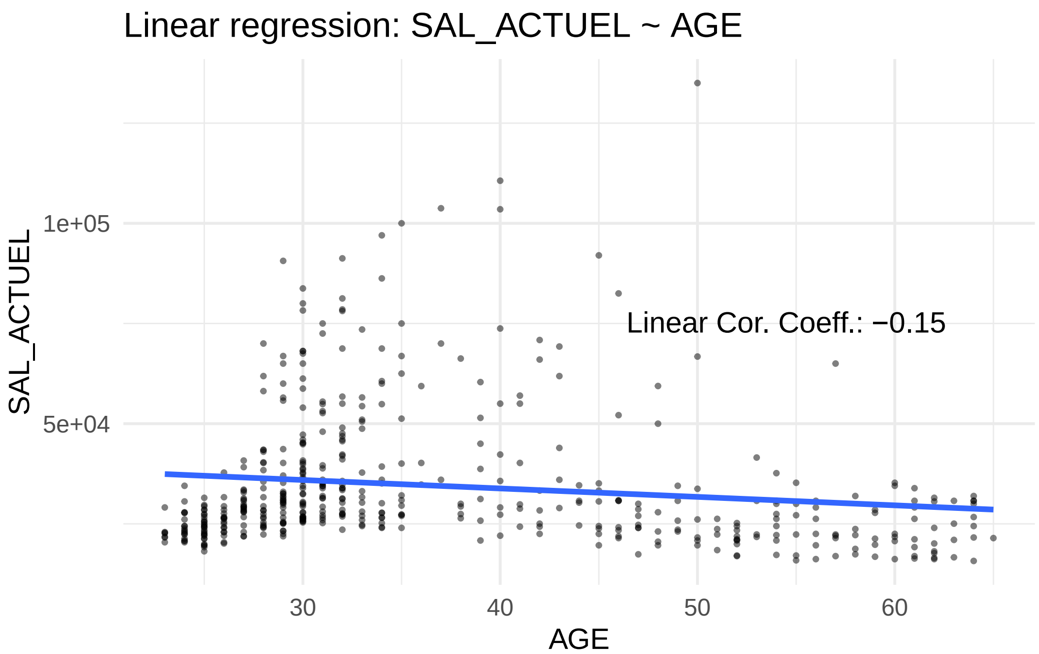


- `ggplot` programming : write a function with arguments `df`, `varx` and `vary` where `varx` and `vary` are two strings denoting numerical columns in `df`, that outputs a ggplot object made of a scatterplot of columns `vary` and `vary`, a linear regression of `vary` against `varx`. The ggplot plot object should be annotated with the linear correlation coefficient of `vary` and `varx` and equipped with a title.

```
bank %>%
  ggplot() +
  aes(x=AGE, y=SAL_ACTUEL) +
  geom_point(alpha=.5, size=.5, ) +
  geom_smooth(method="lm", formula= y ~ x, se=F) +
  annotate(geom="text", x=60, y=60000,
           label=str_c("rho = ",
                       round(cor(bank$SAL_ACTUEL, bank$AGE), 2))) +
  ggtitle("Bank dataset")
```




```r
ggplot_lin_reg <-  function(df, varx, vary){
  rho <- round(cor(df[[varx]], df[[vary]]), 2)
  posx <- sum(range(df[[varx]])*c(.25 , .75))
  posy <- sum(range(df[[vary]])*c(.5 , .5))

  df %>%
    ggplot() +
    aes(x=.data[[varx]], y=.data[[vary]]) +
    geom_point(alpha=.5, size=.5, ) +
    geom_smooth(method="lm", formula= y ~ x, se=F) +
    annotate(geom="text", x=posx, y=posy,
           label=glue("Linear Cor. Coeff.: {rho}")) +
    ggtitle(glue("Linear regression: {vary} ~ {varx}"))
}

ggplot_lin_reg(bank, "AGE", "SAL_ACTUEL")
```



Inspect rows with high Cook's distance

```
lm_1_aug %>%
  filter(.cooksd> 2*mean(.cooksd)) %>%
  select(-starts_with(".")) %>%
  DT::datatable()
```

Show 10 entries Search:

	SEXE	AGE	CATEGORIE	NB_ETUDES	EXPERIENCE	ANCIENNETE	SAL_EMBAUCHE	SAL_ACTUEL
1	0	44	1	16	149	82	27750	34620
2	0	32	5	19	27	64	33000	47550
3	0	48	5	16	264	77	32490	50000
4	0	39	5	18	149	78	36240	51450
5	0	40	5	19	125	65	34980	55000
6	0	43	5	19	26	80	36750	61875
7	0	42	5	16	150	86	47490	66000
8	0	50	7	16	258	83	52500	66750
9	0	43	7	20	134	85	42480	69250
10	0	28	4	16	19	65	21750	70000

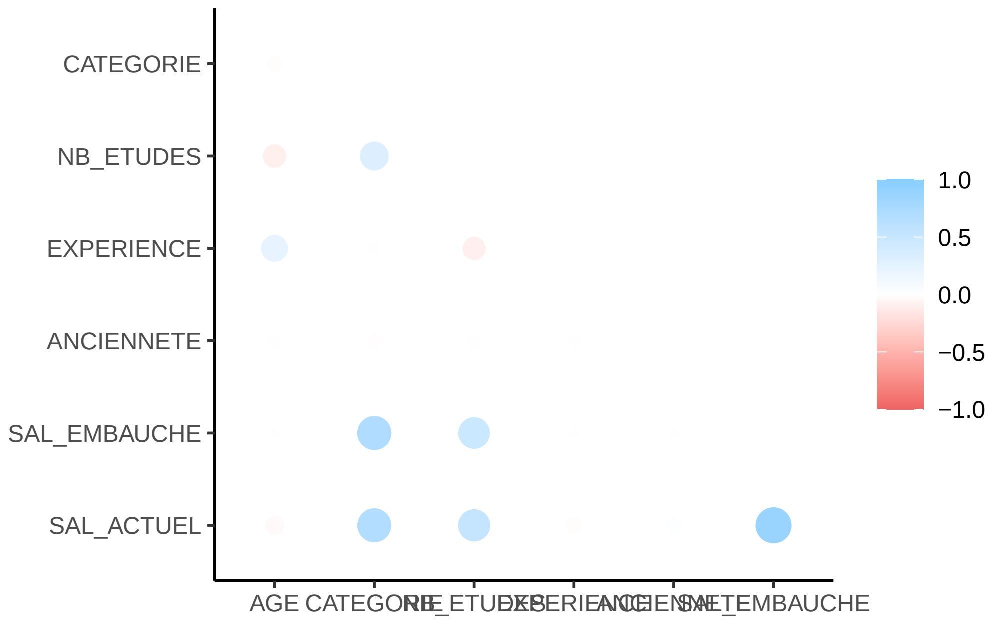
Showing 1 to 10 of 31 entries Previous 1 2 3 4 Next

Discuss the relevance of Simple Linear Regression for analyzing the connection between `SAL_ACTUEL` and `AGE`

Compute the Pearson correlation coefficient for every pair of quantitative variable? Draw corresponding scatterplots.

```
bank %>%
#  select(-id) %>%
  select(where(is.numeric)) %>%
  corrr::correlate() %>%
  corrr::shave() %>%
  corrr::rplot()
```

```
Correlation computed with
* Method: 'pearson'
* Missing treated using: 'pairwise.complete.obs'
```



Predictive linear regression of `SAL_ACTUEL` as a function of age `AGE`

To perform linear fitting, we choose 450 points amongst the 474 sample points: the 24 remaining points are used to assess the merits of the linear fit.

Randomly select 450 rows in the `banque` dataframe.

Function `sample` from base `R` is convenient. You may also enjoy `slice_sample()` from `dplyr`. Denote by `trainset` the vector of of selected indices. Bind the vector of left behind indices to variable `testset`. Functions `match`, `setdiff` or operator `%in%` may be useful.

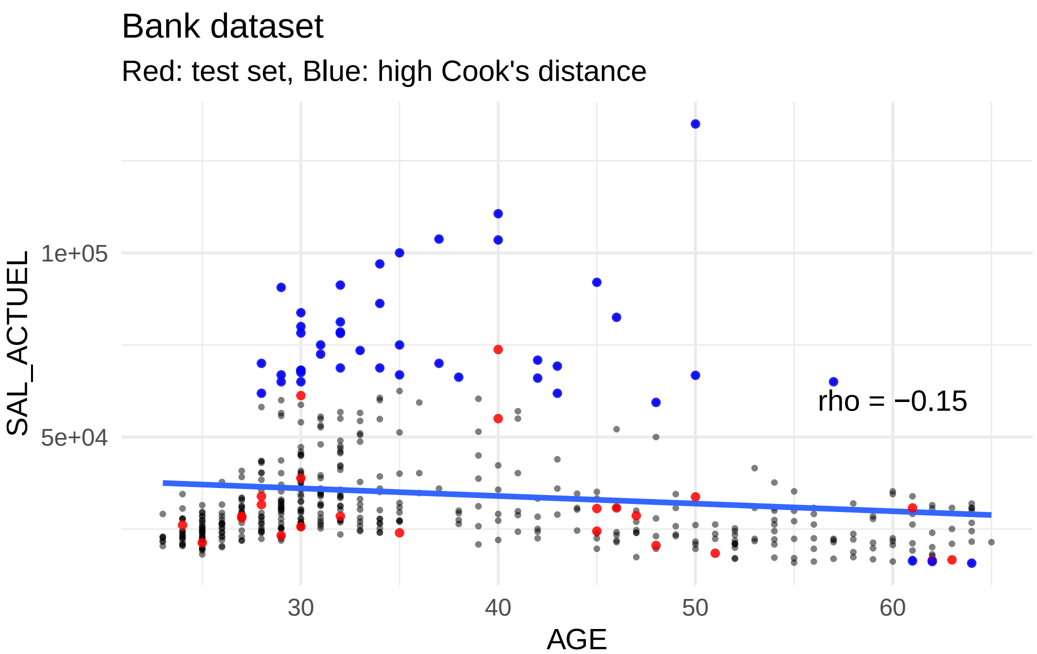
```
old_seed <- set.seed(42)

trainset_size <-  450

trainset <- sample(seq(nrow(bank)) , trainset_size)

testset <- setdiff(seq(nrow(bank)) , trainset)

trainset <- as.integer(trainset)
testset <- as.integer(testset)
```

- ☐ Linear fit of `SAL_ACTUEL` with respect to `AGE`, on the training set. Call the result `lm_3`.
- ☐ How do you feel about such a linear fit? (Use diagnostic plots)

```
lm_3 <- lm(SAL_ACTUEL ~ AGE, data=bank[trainset,] )
#
lm_3_aug <-  lm_3 %>%
  augment(data=bank[trainset,] )
```

```
lm_3 %>%
  augment(data=bank[trainset,]) %>%
  ggplot() +
  aes(x=AGE, y=SAL_ACTUEL) +
  geom_point(alpha=.5, size=.5) +
```

```
  geom_smooth(method="lm", formula= y ~ x, se=F) +
  annotate(geom="text", x=60, y=60000,
           label=str_c("rho = ",
                       round(cor(bank$SAL_ACTUEL, bank$AGE), 2))) +
  ggtitle("Bank dataset",
          subtitle = "Red: test set, Blue: high Cook's distance") +
  geom_point(data=augment(lm_3, newdata=bank[testset,]),
             color="red",
             alpha=.85, size=1) +
  geom_point(data=filter(lm_3_aug, .cooksd> 2*mean(.cooksd)),
             color="blue",
             alpha=.85, size=1
             )
```



Inspecting points with high Cook's distance

```
lm_3_aug %>%
  filter(.cooksd> 2*mean(.cooksd)) %>%
  select(-starts_with(".")) %>%
  DT::datatable()
```

Show 10 entries Search:

	SEXE	AGE	CATEGORIE	NB_ETUDES	EXPERIENCE	ANCIENNETE	SAL_EMBAUCHE	SAL_ACTUEL
1	0	50	7	19	199	96	79980	135000
2	0	30	2	15	34	82	15750	80000
3	1	62	1	12	180	68	10200	16200
4	0	30	4	16	12	79	21750	83750
5	0	29	1	18	30	79	31980	66875
6	0	34	5	19	68	91	35010	97000
7	0	43	7	20	134	85	42480	69250
8	0	30	4	19	29	78	32010	68125
9	0	35	6	19	13	65	42510	75000
10	0	34	4	17	8	89	27510	68750

Showing 1 to 10 of 44 entries Previous 1 2 3 4 5 Next

```
make_p_diag_1(lm_3) +
make_p_diag_2(lm_3) +
make_p_diag_3(lm_3) +
make_p_diag_5(lm_3)
```

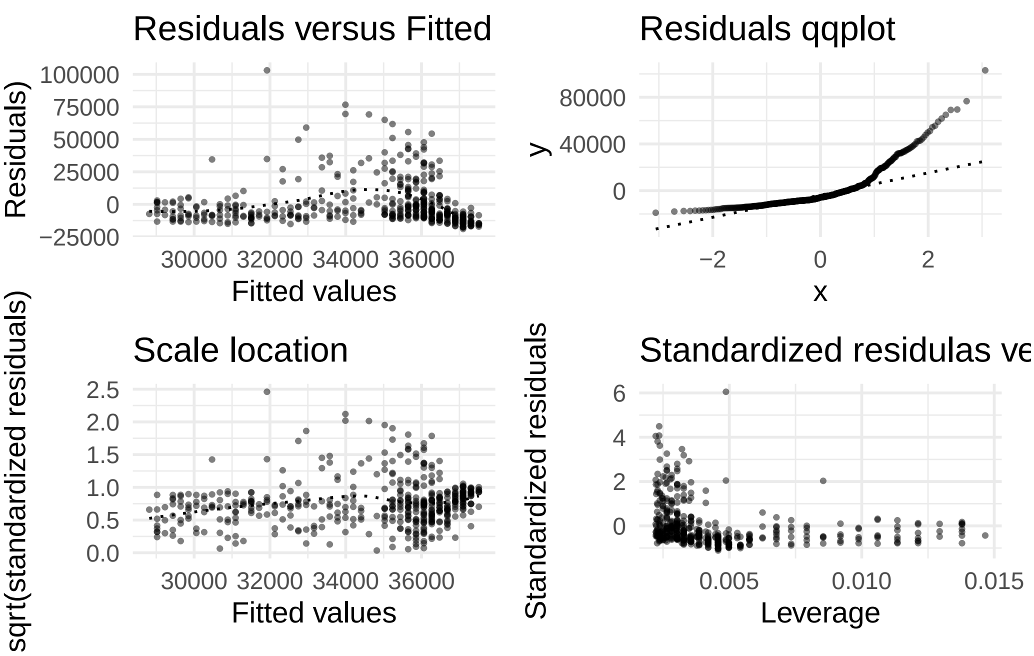


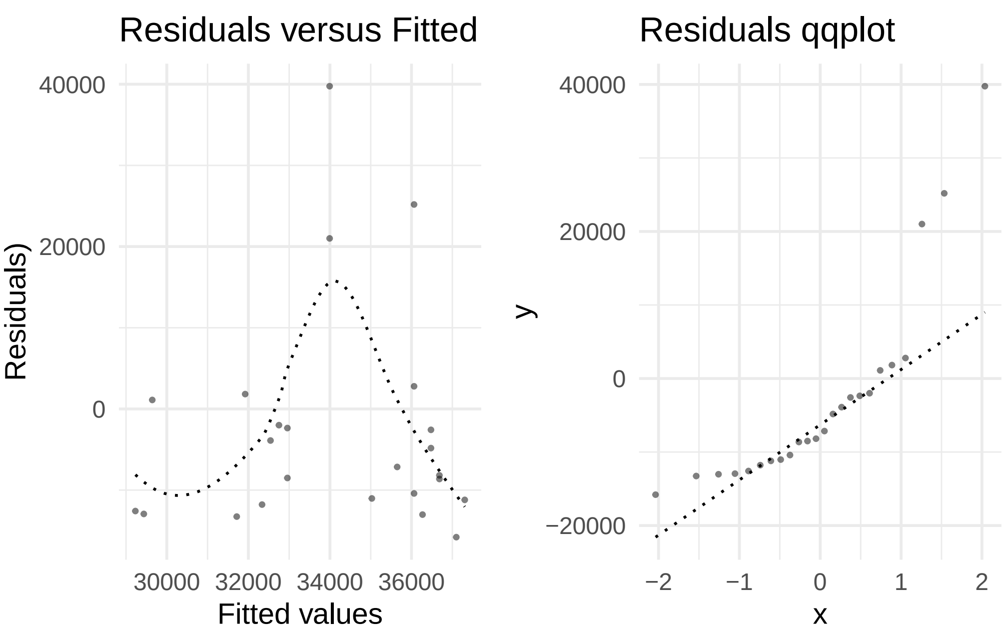
Figure 1: SAL_ACTUEL ~ AGE on training set

- ☐ Use `lm_3` to predict the values of `SAL_ACTUEL` as an affine function of `AGE` on the testing set `testset` (`broom::augment()` with optional argument `newdata` may be useful). Compare the data frame with the one obtained from `augment(lm_3)`.

```
lm_3_aug_test <- augment(lm_3, newdata = bank[testset,])
```

- ☐ Compare training error and testing error
- ☐ Analyse residuals (prediction errors) on the testing set. Compare with training set

```
(make_p_diag_1(lm_3) %+% lm_3_aug_test) +
(make_p_diag_2(lm_3) %+% lm_3_aug_test)
```



Expectations under Gaussian Linear Modelling Assumptions

$$(Y) = (\mathbb{Z}) \times \beta + \sigma\,(\epsilon)$$

```r
old_seed <- set.seed(5783)

# lm_1 %>%
#   tidy()

#lm_1 %>% summary
```

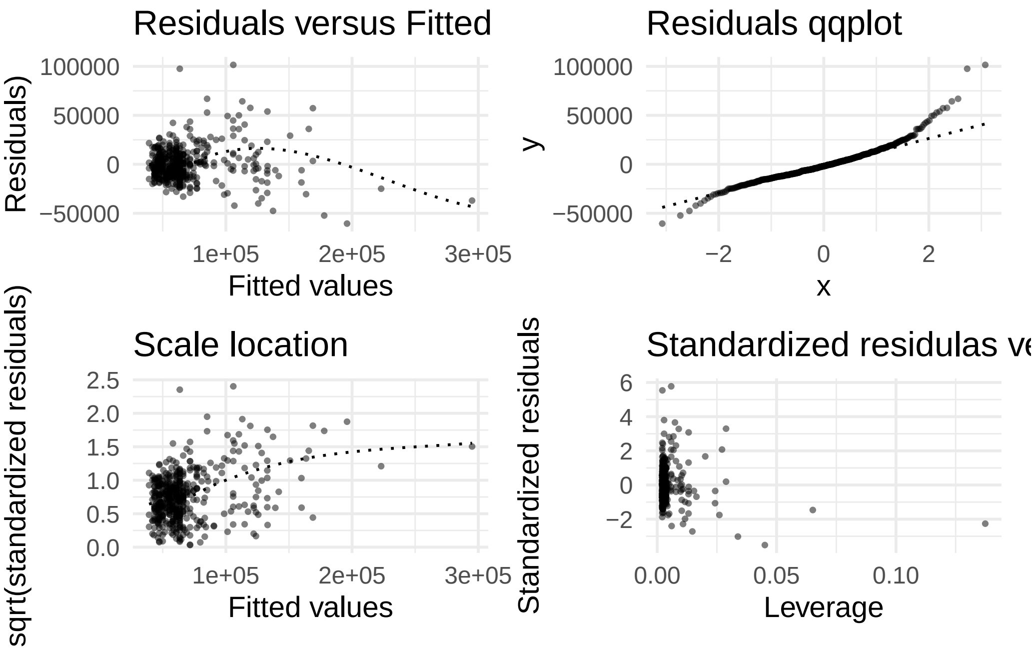
```r
lm2design <- . %$%      # exposing pipe from magrittr
  select(.$model, -ncol(.$model)) %>%
  mutate(ctt = 1) %>%
  select(ctt, everything()) %>%
  as.matrix() # design matrix
```

```r
sigma_hat <- sqrt(sum(lm_1$residuals^2)/lm_1$df.residual)

sal_actuel_fake <- lm2design(lm_1) %*% lm_1$coefficients +
  sigma_hat * rnorm(nrow(lm_1$model))

lm_1_fake <- bind_cols(bank,
                       SAL_ACTUEL_FAKE= sal_actuel_fake) %>%
  lm(formula=SAL_ACTUEL_FAKE ~ SAL_EMBAUCHE, data=.)
```

```r
summary(lm_1_fake)
```

```
Call:
lm(formula = SAL_ACTUEL_FAKE ~ SAL_EMBAUCHE, data = .)

Residuals:
   Min     1Q Median     3Q    Max
-60551 -10758  -1974   7942 101530

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  6722.120   1930.957   3.481 0.000545 ***
SAL_EMBAUCHE    3.606      0.103  35.003  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 17630 on 472 degrees of freedom
Multiple R-squared:  0.7219,    Adjusted R-squared:  0.7213
F-statistic:  1225 on 1 and 472 DF,  p-value: < 2.2e-16
```

```r
#
make_p_diag_1(lm_1_fake) +
make_p_diag_2(lm_1_fake) +
make_p_diag_3(lm_1_fake) +
make_p_diag_5(lm_1_fake)
```


Residuals versus Fitted
Residuals)
100000
50000
0
−50000
1e+05
2e+05
3e+05
Fitted values
Residuals qqplot
y
100000
50000
0
−50000
−2
0
2
x
Scale location
sqrt(standardized residuals)
2.5
2.0
1.5
1.0
0.5
0.0
1e+05
2e+05
3e+05
Fitted values
Standardized residulas ve
Standardized residuals
6
4
2
0
−2
0.00
0.05
0.10
Leverage