Linear regression I

2024-09-05

ToC

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

> **!** **Objectives**

Setup

```
stopifnot(
  require(broom),
  require(corrr),
  require(DT),
  require(GGally),
  require(ggforce),
  require(glue),
  require(gt),
  require(httr),
  require(kableExtra),
  require(lobstr),
  require(magrittr),
  require(patchwork),
  require(rlang),
```

```r
  require(skimr),
  require(fs),
  require(tidyverse),
  require(viridis)
)

old_theme <- theme_set(theme_minimal())

knitr::opts_chunk$set(
  message = FALSE,
  warning = FALSE,
  comment=NA,
  prompt=FALSE,
  cache=FALSE,
  echo=TRUE,
  results='asis'
)
```

Dataset

Dataset `Banque.csv` contains information on clerical officers in the Banking sector. We aim at investigating connections between variables.

Per-column analysis

- ☐ Load the dataset.
- ☐ Have a glimpse at it
- ☐ Check the typing of columns

```r
if (fs::dir_exists('DATA')){
  datapath <- '/DATA'
} else {
  datapath <- '../DATA'
}
```

```r
fpath <- str_c(datapath, "Banque.csv", sep="/")  # tune this

bank <- readr::read_delim(fpath, delim = "\t",
    escape_double = FALSE,
    col_types = cols(
        SEXE = col_character(),
        CATEGORIE = col_character(),
        NB_ETUDES = col_character()),
    trim_ws = TRUE)

# View(bank)

bank %>%
  glimpse(withd=50)
```

```
Rows: 474
Columns: 14
$ SEXE            <chr> "1", "1", "1", "1", "1", "1", "1", "1", "1", "1", "1",~
```

```
$ AGE             <dbl> 64, 55, 56, 60, 62, 61, 62, 63, 59, 61, 52, 57, 55, 52~
$ CATEGORIE       <chr> "1", "1", "1", "1", "1", "1", "1", "1", "1", "1", "1",~
$ NB_ETUDES       <chr> "8", "8", "8", "12", "12", "12", "12", "8", "8", "12",~
$ EXPERIENCE      <dbl> 275, 43, 0, 0, 180, 163, 288, 412, 76, 124, 271, 319, ~
$ ANCIENNETE      <dbl> 70, 74, 92, 82, 68, 66, 84, 88, 76, 97, 72, 72, 85, 81~
$ SAL_EMBAUCHE    <dbl> 10200, 10200, 9750, 10200, 10200, 10200, 10200, 9750, ~
$ SAL_ACTUEL      <dbl> 15750, 15900, 16200, 16200, 16200, 16350, 16500, 16650~
$ VILLE           <chr> "PARIS", "PARIS", "LILLE", "BORDEAUX", "PARIS", "PARIS~
$ SATIS_EMPLOI    <chr> "non", "non", "oui", "non", "non", "non", "oui", "non"~
$ SATIS_CHEF      <chr> "oui", "oui", "non", "non", "oui", "oui", "non", "non"~
$ SATIS_SALAIRE   <chr> "non", "non", "non", "non", "oui", "non", "non", "non"~
$ SATIS_COLLEGUES <chr> "non", "non", "oui", "oui", "oui", "non", "non", "oui"~
$ SATIS_CE        <chr> "oui", "oui", "oui", "oui", "oui", "oui", "oui", "oui"~
```

```
The table schema is the following

-   SEXE :
    -   "0" : Man,
    -   "1" : Woman.
-   AGE : in years.
-   CATEGORIE : Employment category (from 1 to 7).
-   NB_ETUDES : Number of years of education
-   EXPERIENCE : Previous Expérience antérieure (in months).
-   ANCIENNETE : Seniority in this bank (in months).
-   SAL_EMBAUCHE : Starting salary (Euros).
-   SAL_ACTUEL : Present salary (Euros).
-   VILLE : City of residence
-   SATIS_EMPLOI : Satisfied with your job?
-   SATIS_CHEF : Satisfied with your manager?
-   SATIS_SALAIRE : Satisfied with your salary?
-   SATIS_COLLEGUES : Satisfied with your colleagues?
-   SATIS_CE : Happy with your works council?
```

☐ Define population and individuals.

```
- Population: Bank employees in France
- Sample: Those employees who answered the questionnaire (we have no clues about possible
```

☐ Determine the type and domain of each variable.

```r
make_biotifoul <-  function(df, .f=is.factor){
  .scales <- ifelse(identical(.f, is.factor), "free_x", "free")

  p <- df %>%
    select(where(.f)) %>%
    pivot_longer(
      cols = everything(),
      names_to = "var",
      values_to = "val"
    ) %>%
    ggplot() +
    aes(x = val) +
    facet_wrap(~var, scales=.scales) + xlab("")

  if(identical(.f, is.factor)){
```

```
    p + geom_bar()
  } else {
    p + geom_histogram(aes(y=after_stat(density)), bins=30) + xlab("")
  }
}
```

Preliminary inspection.

```
bank %>%
  skim() %>%
  DT::datatable(extensions=c("Responsive"))
```

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

Show 10 entries Search:

	skim_type	skim_variable	n_missing	complete_rate	character.min	character.max	character.empty	character.
▶ 1	character	SEXE	0	1	1	1	0	
▶ 2	character	CATEGORIE	0	1	1	1	0	
▶ 3	character	NB_ETUDES	0	1	1	2	0	
▶ 4	character	VILLE	0	1	4	9	0	
▶ 5	character	SATIS_EMPLOI	0	1	3	3	0	
▶ 6	character	SATIS_CHEF	0	1	3	3	0	
▶ 7	character	SATIS_SALAIRE	0	1	3	3	0	
▶ 8	character	SATIS_COLLEGUES	0	1	3	3	0	
▶ 9	character	SATIS_CE	0	1	3	3	0	
▶ 10	numeric	AGE	0	1				

Showing 1 to 10 of 14 entries Previous 1 2 Next

- ☐ Make all columns with less than 10 distinct values `factor` or `logical`.

All columns with less than 10 distinct values but more than 2 values will be considered as factors.

```
to_factorize <- bank %>%
  summarise(across(everything(), n_distinct)) %>%
  pivot_longer(cols=everything(),
               names_to="name",
               values_to = "n_distinct") %>%
  filter(n_distinct<= 10) %>%
  pull("name")
```

```
bank <- bank %>%
  mutate(
    across(starts_with('SATIS_'),     # tidy selection
           ~ ifelse(.=="oui", TRUE, FALSE))
  )

bank <- bank %>%
  mutate(SEXE=ifelse(SEXE==1, "F", "M"))
```

```r
to_factorize <- bank %>%
  select(-where(is_logical)) %>%   # tidy selection
  summarise(across(everything(), n_distinct)) %>%  # tidy selection
  pivot_longer(everything(),
               names_to = "col_name",
               values_to = "n_distinct") %>%
  filter(n_distinct<=10) %>%
  pluck("col_name")

bank <- bank %>%
  mutate(across(all_of(to_factorize), as_factor))  # tidy selection
```

```r
bank <- bank %>%
  mutate(SEXE= fct_recode(SEXE, "M"="0", "F"="1"))
```

```
Warning: There was 1 warning in `mutate()`.
i In argument: `SEXE = fct_recode(SEXE, M = "0", F = "1")`.
Caused by warning:
! Unknown levels in `f`: 0, 1
```

It looks better!

```r
bank %>%
  skim(where(is.numeric)) %>%
  DT::datatable(extensions = c("Responsive"))
```

Show 10 entries Search:

	skim_type	skim_variable	n_missing	complete_rate	numeric.mean	numeric.sd	numeric.p0	numeric.p2
▶ 1	numeric	AGE	0	1	37.23839662447257	11.81589401336001	23	
▶ 2	numeric	EXPERIENCE	0	1	95.86075949367088	104.5862361045115	0	1[illegible]
▶ 3	numeric	ANCIENNETE	0	1	81.1097046413502	10.06094487371352	63	
▶ 4	numeric	SAL_EMBAUCHE	0	1	17016.08649789029	7870.638154474875	9000	124[illegible]
▶ 5	numeric	SAL_ACTUEL	0	1	34419.56751054852	17075.66146458606	15750	24

Showing 1 to 5 of 5 entries Previous 1 Next

```r
bank %>%
  skim(where(is.factor)) %>%
  DT::datatable(extensions = c("Responsive"))
```

Show 10 entries Search:

	skim_type	skim_variable	n_missing	complete_rate	factor.ordered	factor.n_unique	factor.top_counts
1	factor	SEXE	0	1	false	2	M: 258, F: 216
2	factor	CATEGORIE	0	1	false	7	1: 227, 2: 136, 4: 41, 5: 32
3	factor	NB_ETUDES	0	1	false	10	12: 190, 15: 116, 16: 59, 8: 53
4	factor	VILLE	0	1	false	7	PAR: 92, LIL: 84, BOR: 81, LYO: 81

Showing 1 to 4 of 4 entries Previous 1 Next

```
bank %>%
  skim(where(is.logical)) %>%
  DT::datatable(extensions = c("Responsive"))
```

Show 10 entries Search:

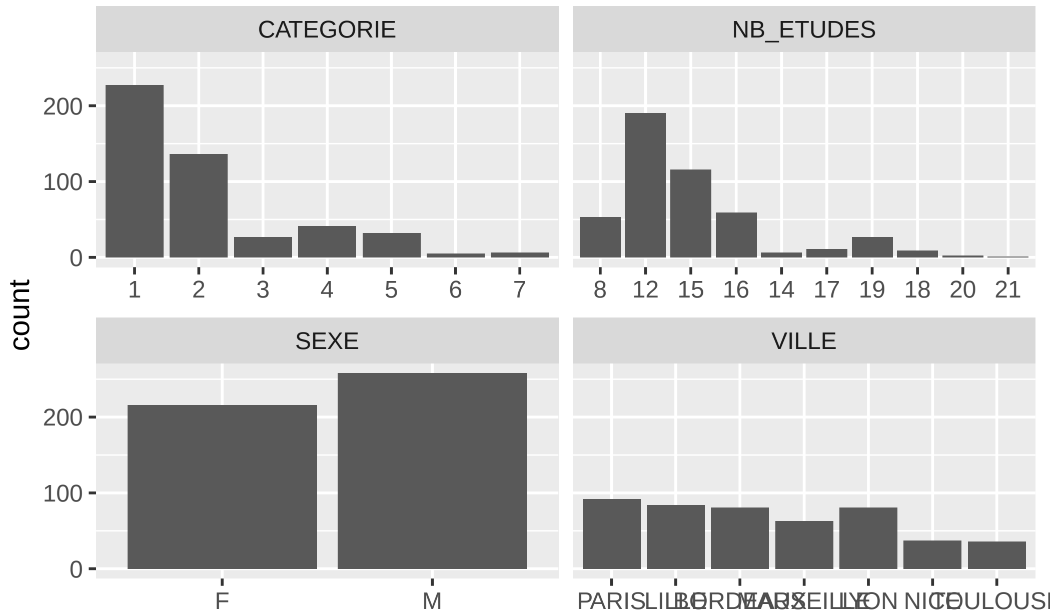
	skim_type	skim_variable	n_missing	complete_rate	logical.mean	logical.count
1	logical	SATIS_EMPLOI	0	1	0.4050632911392405	FAL: 282, TRU: 192
2	logical	SATIS_CHEF	0	1	0.6139240506329114	TRU: 291, FAL: 183
3	logical	SATIS_SALAIRE	0	1	0.1012658227848101	FAL: 426, TRU: 48
4	logical	SATIS_COLLEGUES	0	1	0.3354430379746836	FAL: 315, TRU: 159
5	logical	SATIS_CE	0	1	0.8270042194092827	TRU: 392, FAL: 82

Showing 1 to 5 of 5 entries Previous 1 Next

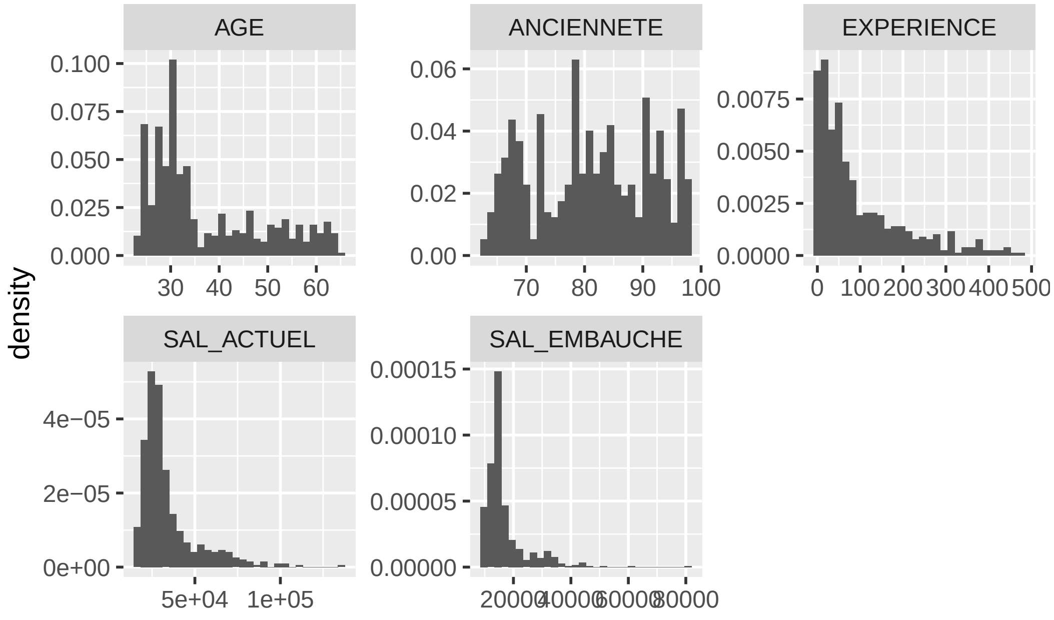
We add an identifier column so as to identify rows

```
bank <- bank %>%
  rownames_to_column(var="id")
```

```
bank %>%
  select(-id) %>%
  make_biotifoul(.f=is.factor)
```




```
bank %>%
  select(-id) %>%
  make_biotifoul(.f=is.numeric)
```



Pairwise scan

Use `pairs()` of `ggpairs()` to scan pairwise interactions between columns

`ggpairs()` explores all pairwise interactions. It is time-consuming.

```
# bank %>%
#   ggpairs()
```

Function `pairs` works with numerical columns

```
bank %>%
  select(where(is.numeric)) %>%
  pairs()
```

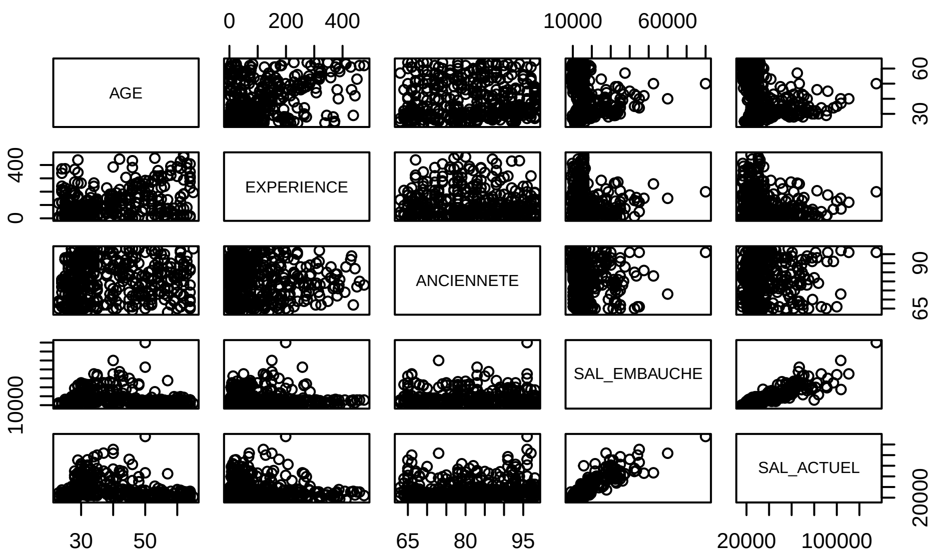


Figure 1: Pairwise interactions between numerical columns of bank dataset

As we intend to *explain* `SAL_ACTUEL` as a function of the other variables, the last row is interesting. `SAL_EMBAUCHE` looks more promising than the three other covariates.

```
bank %>%
  select(where(is.numeric)) %>%
  pivot_longer(cols=-c("SAL_ACTUEL"),
               names_to="covariate",
               values_to = "X")  %>%
  ggplot() +
    aes(y=SAL_ACTUEL, x=X) +
    geom_point(alpha=.5, size=.5) +
    facet_wrap(~ covariate, scales = "free_x") +
    xlab("") +
  ggtitle("SAL_ACTUEL versus other numerical covariates",
          subtitle="Banque dataset")
```

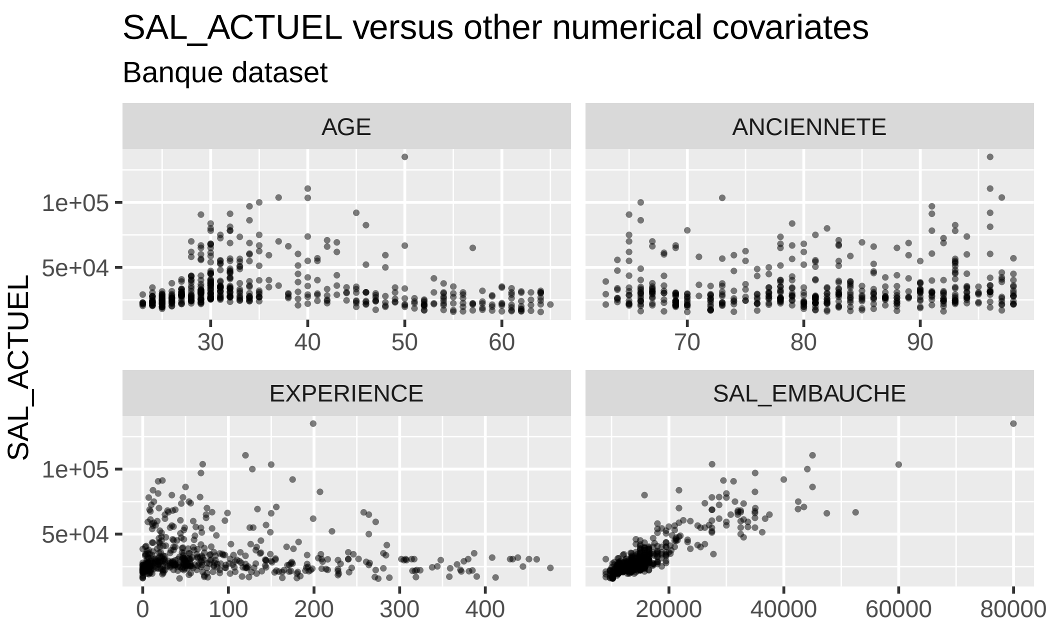


Figure 2: Pairwise interactions between numerical covariates and and response variable SAL_ACTUEL

Linear correlation coefficient

We first investigate connexion between salary at hiring time (`SAL_EMBAUCHE`) and current salary (*SAL_ACTUEL*).

☐ Redraw a scatterplot. Observations?

```
p_scat <- bank %>%
  ggplot() +
  aes(x=SAL_EMBAUCHE, y=SAL_ACTUEL) +
  geom_point(alpha=.5, size=.5 ) +
#  geom_jitter(alpha=.25, size=.5)
  ggtitle("Bank dataset")
```

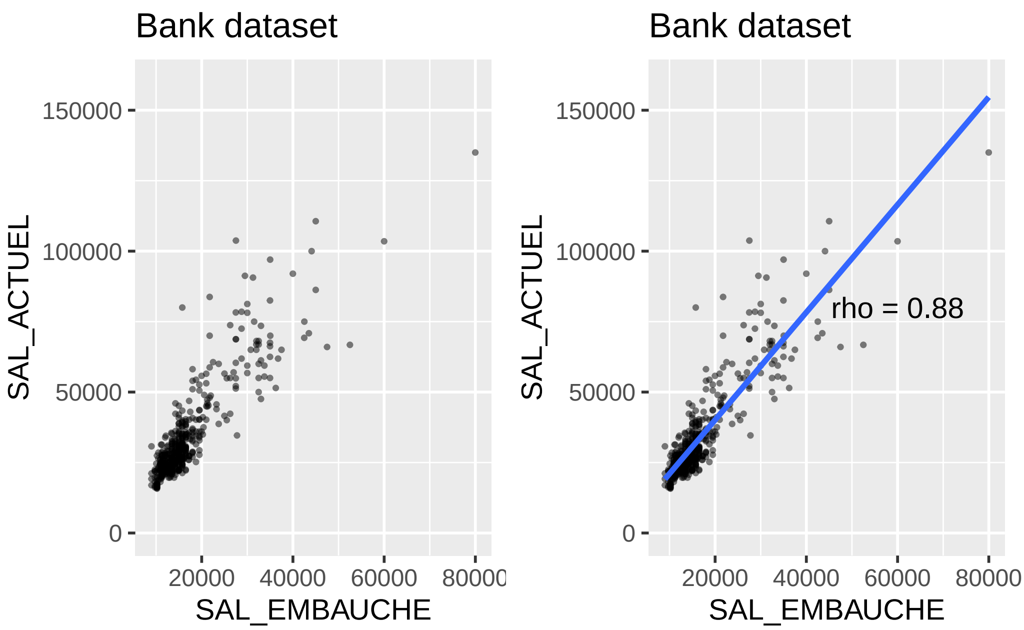
☐ Compute the Pearson correlation coefficient (using `cor`). Recall the formal definition of Pearson's correlation coefficient;

☐ Redraw the scatterplot and overlay it with a regression line.

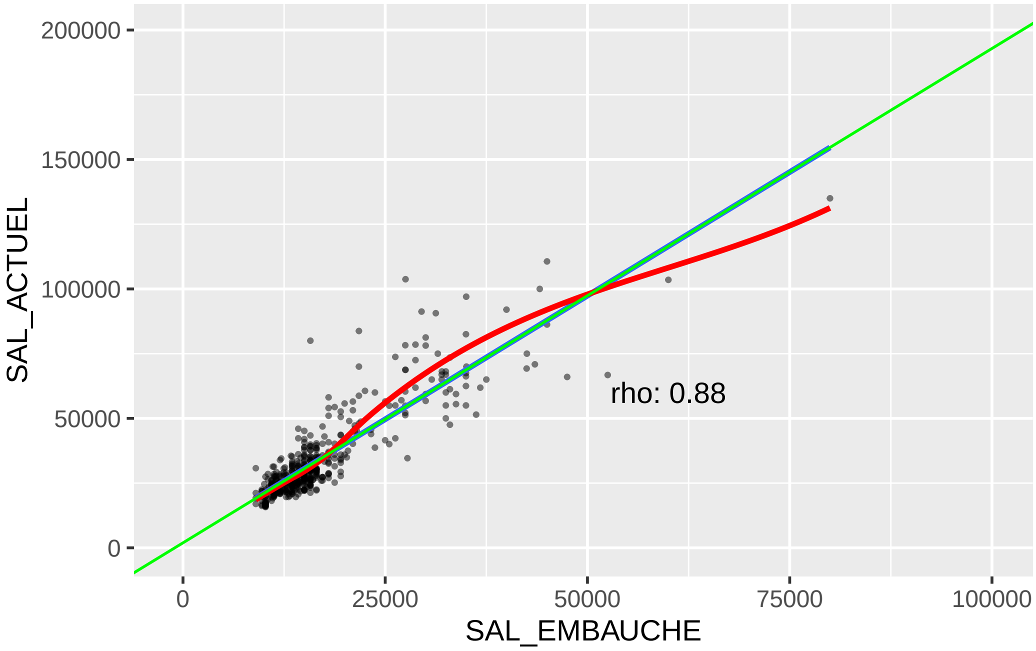
☐ Conclusion ?

Whith correlation coefficient.

```
rho <- cor(bank$SAL_ACTUEL, bank$SAL_EMBAUCHE)

p_scat_reg_lin <- p_scat +
  geom_smooth(method="lm", formula= y ~ x, se=F) +
  annotate(geom="text",
           x=60000, y=80000,
           label=glue("rho = {round(rho, 2)}"))

(p_scat + ylim(c(0,160000))) + (p_scat_reg_lin + ylim(c(0,160000)))
```

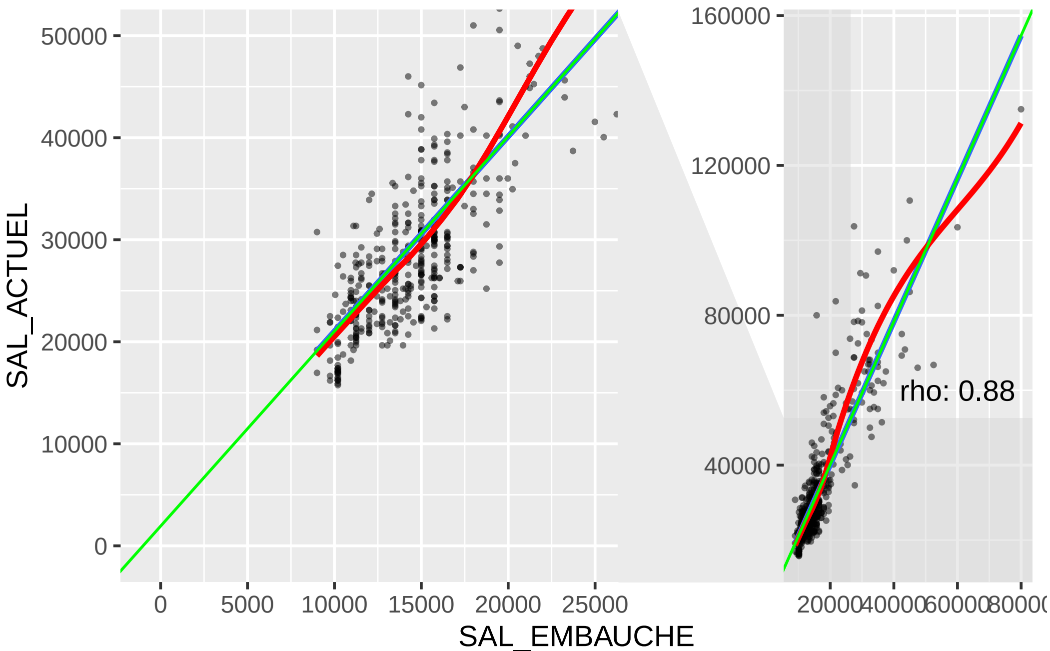



```
rho <- cor(bank$SAL_ACTUEL, bank$SAL_EMBAUCHE)

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

p_scat_1 <- bank %>%
  ggplot() +
  aes(x=SAL_EMBAUCHE, y= SAL_ACTUEL) +
  geom_point(alpha=.5, size=.5) +
  annotate(geom="text", x=60000, y=60000, label=str_c("rho: ", round(rho,2))) +
  geom_smooth(method = "lm", formula = y ~ x, se=F) +
  geom_smooth(method= "loess", formula = y ~ x, se=F, color="red") +
  geom_abline(slope=coefficients(lm_1)[2],
              intercept =coefficients(lm_1)[1], color="green" ) # +
#  coord_fixed()

p_scat_1  +
  xlim(c(-1000, 1e5)) + ylim(c(-1000, 2e5))
```



☐ Zoom on low incomes

```
p_scat_1  +
  ggforce::facet_zoom(xlim=c(-1000,25000),ylim=c(-1000,50000))
```



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.

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

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()`

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

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()`

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

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

Table 3: SAL_ACTUEL ~ SAL_EMBAUCHE

id	SEXE	AGE	CATEGORIE	NB_ETUDES	EXPERIENCE	ANCIENNETE	SAL_EMBAUCHE	SAL_ACTUEL	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	.fitted	.resid	.hat	.sigma	.cooksd	.std.resid
1	F	64	1	8	275	70	10200	15750	PARIS	FALSE	TRUE	FALSE	FALSE	TRUE	21404.59	-5654.59	0	8119.77	0	-0.70
2	F	55	1	8	43	74	10200	15900	PARIS	FALSE	TRUE	FALSE	FALSE	TRUE	21404.59	-5504.59	0	8119.99	0	-0.68
3	F	56	1	8	0	92	9750	16200	LILLE	TRUE	FALSE	FALSE	TRUE	TRUE	20545.34	-4345.34	0	8121.49	0	-0.54
4	F	60	1	12	0	82	10200	16200	BORDEAUX	FALSE	FALSE	FALSE	TRUE	TRUE	21404.59	-5204.59	0	8120.41	0	-0.64
5	F	62	1	12	180	68	10200	16200	PARIS	FALSE	TRUE	TRUE	TRUE	TRUE	21404.59	-5204.59	0	8120.41	0	-0.64
6	F	61	1	12	163	66	10200	16350	PARIS	FALSE	TRUE	FALSE	FALSE	TRUE	21404.59	-5054.59	0	8120.61	0	-0.62

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

Table 4: 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.
```

```
# plot(lm.1, which=3)
```

☐ 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) +
```

```r
  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)
```

```r
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

```r
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)     # DRY this ?
```

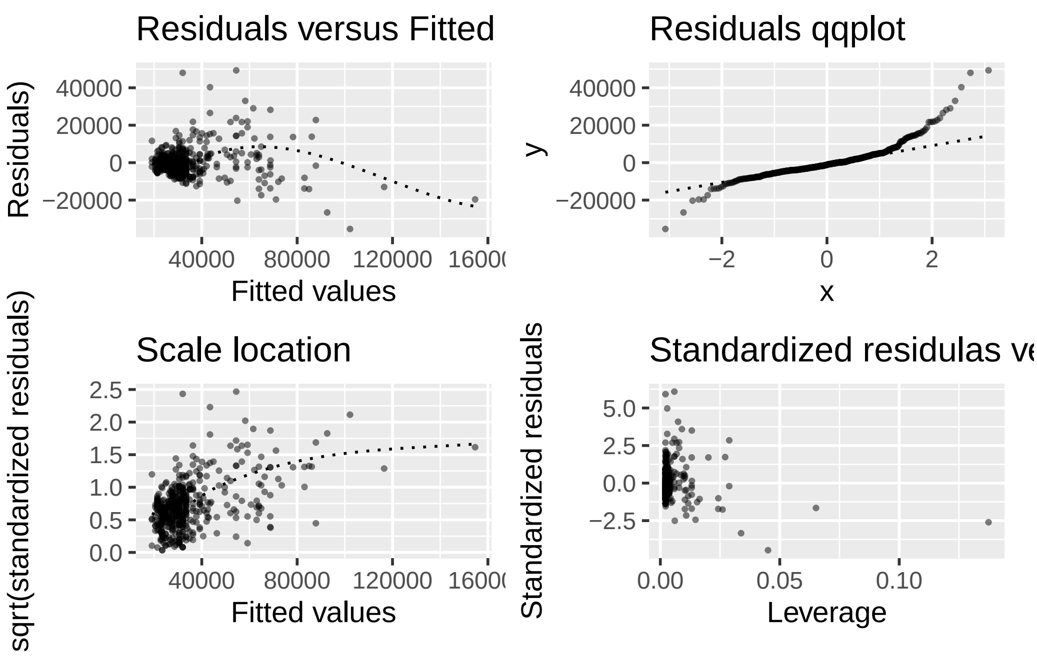
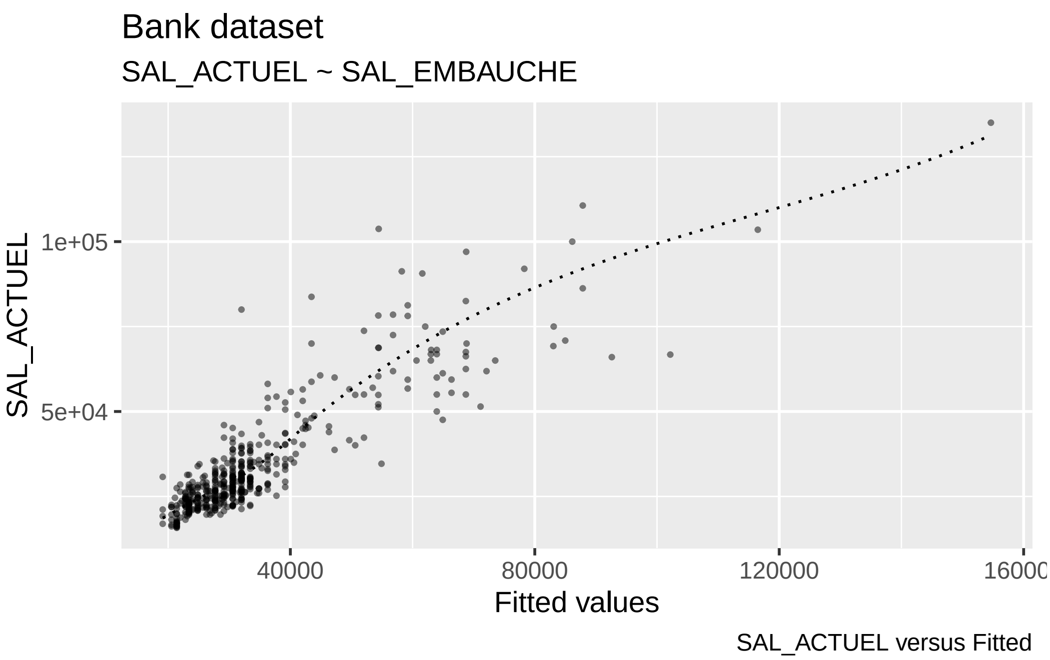


Figure 3: SAL_ACTUEL ~ SAL_EMBAUCHE

```
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)
```

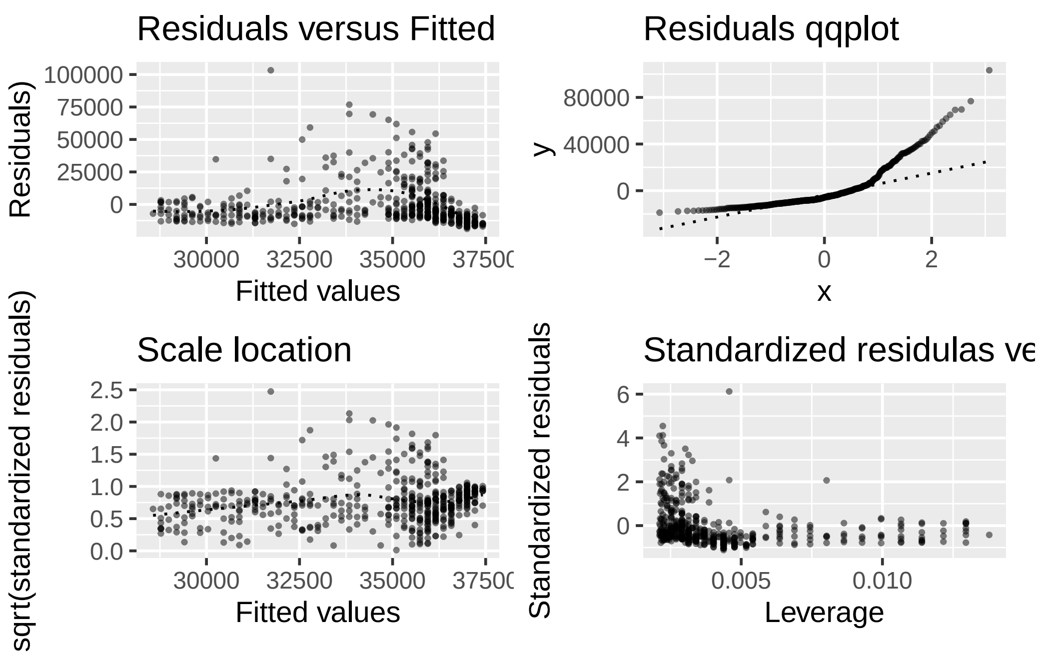
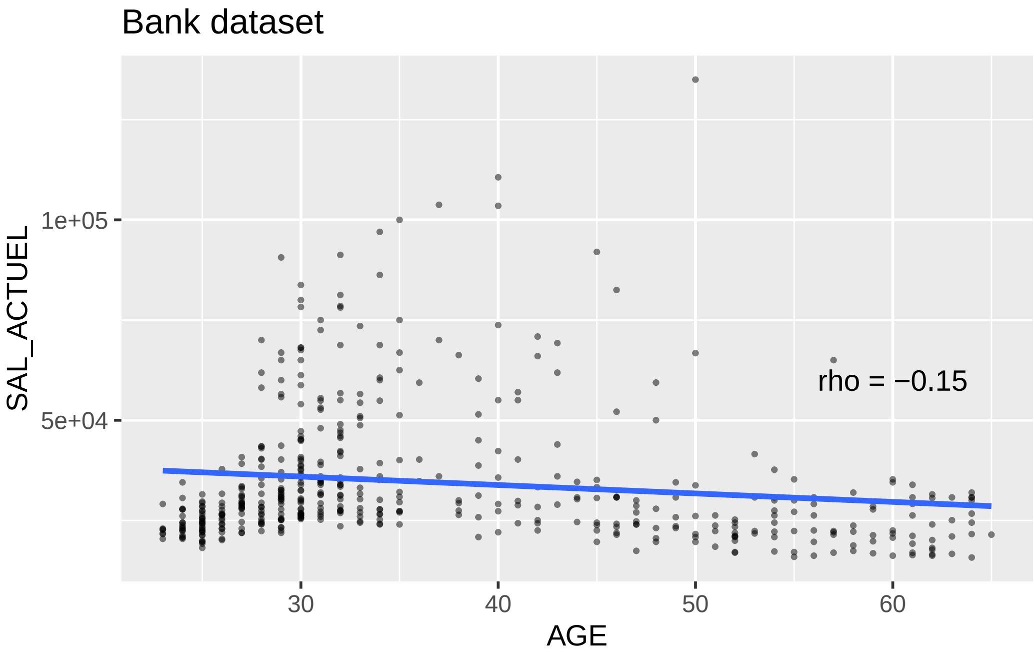


Figure 4: SAL_ACTUEL ~ AGE

```
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")
```



Inspect rows with high Cook's distance

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

Show 10 entries Search:

	id	SEXE	AGE	CATEGORIE	NB_ETUDES	EXPERIENCE	ANCIENNETE	SAL_EMBAUCHE	SAL_ACTU
1	335	M	44	1	16	149	82	27750	
2	399	M	32	5	19	27	64	33000	
3	403	M	48	5	16	264	77	32490	
4	407	M	39	5	18	149	78	36240	
5	417	M	40	5	19	125	65	34980	
6	434	M	43	5	19	26	80	36750	
7	439	M	42	5	16	150	86	47490	
8	441	M	50	7	16	258	83	52500	
9	449	M	43	7	20	134	85	42480	
10	450	M	28	4	16	19	65	21750	

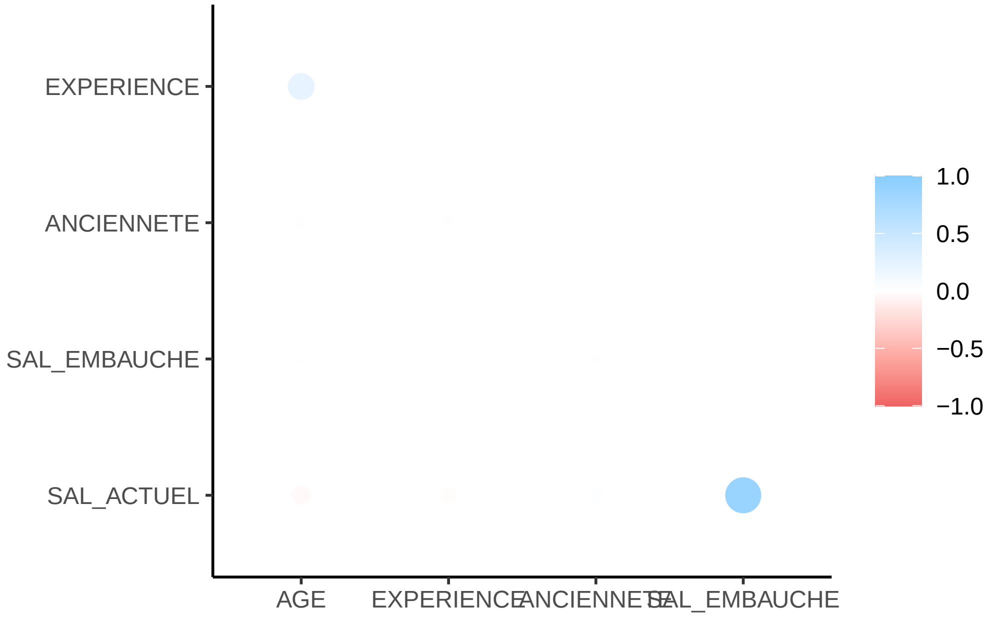
Showing 1 to 10 of 31 entries Previous 1 2 3 4 Next

- ☐ Discuss the relevance of Simple Linear Regression for analyzing the connection between `SAL_ACTUEL` and `SAL_EMBAUCHE`
- ☐ 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.
- ☐ 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)

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

trainset_size <-  450

trainset <- sample(pluck(bank, "id") , trainset_size)

testset <- setdiff(pluck(bank, "id") , trainset)

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

# foo <- slice_sample(bank, n = trainset_size)
```