EY 2025 Case Comp

library(tidyverse)

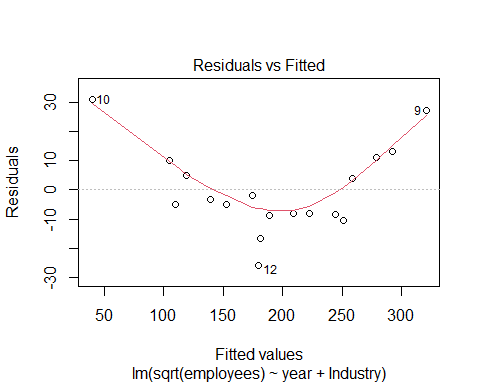
── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
✔ dplyr 1.1.4 ✔ readr 2.1.5  
✔ forcats 1.0.0 ✔ stringr 1.5.1  
✔ ggplot2 3.5.1 ✔ tibble 3.2.1  
✔ lubridate 1.9.4 ✔ tidyr 1.3.1  
✔ purrr 1.0.2   
── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
✖ dplyr::filter() masks stats::filter()  
✖ dplyr::lag() masks stats::lag()  
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

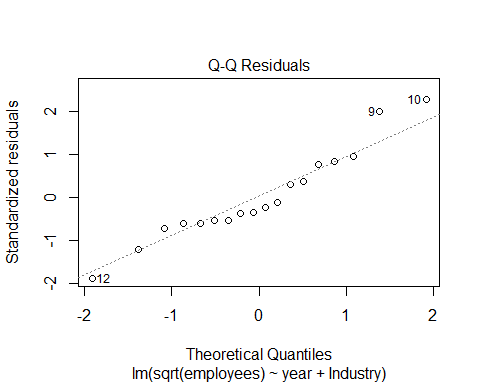
library(readxl)  
library(patchwork)  
data <- read\_excel("2025 Case Comp Data CLEAN.xlsx")  
  
# Workforce size by industry ----  
employees\_table <- data %>%   
 select(`Company ID`, Industry, `2023` = `Number of employees 2023`, `2024` = `Number of employees 2024`, `2025` = `Number of employees 2025`) %>%   
 pivot\_longer(cols = 3:5, names\_to = "year", values\_to = "employees") %>%  
 mutate(year = as.numeric(year)) %>%   
 group\_by(Industry, year) %>%   
 summarise(employees = sum(employees))

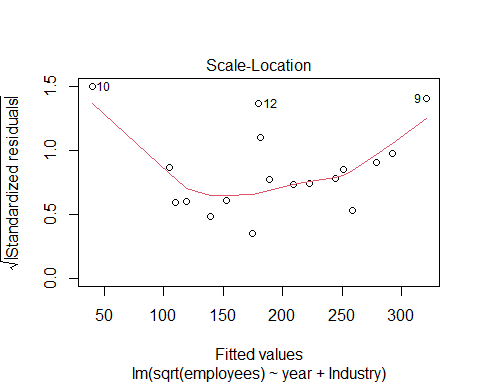
`summarise()` has grouped output by 'Industry'. You can override using the  
`.groups` argument.

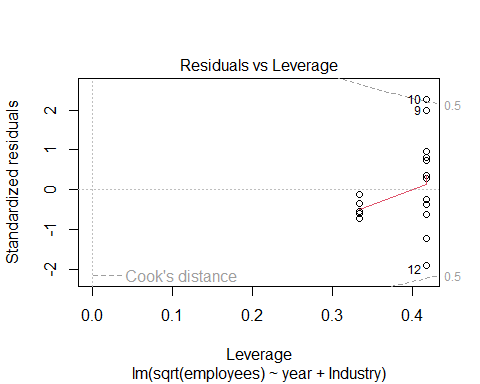
predict\_employees <- function(industry, summary = TRUE, assumption = TRUE) {  
 model <- lm(sqrt(employees) ~ year + Industry, data = employees\_table)  
 if(summary) {print(summary(model))}  
 if(assumption) {print(plot(model))}  
 predicted\_value <- predict(model, newdata = data.frame(year = 2026, Industry = industry))  
 # result <- list(model = model, prediction = predicted\_value)  
 # return(result)  
 return(predicted\_value^2)  
}  
  
predict\_employees("Education")

Call:  
lm(formula = sqrt(employees) ~ year + Industry, data = employees\_table)  
  
Residuals:  
 Min 1Q Median 3Q Max   
-25.747 -8.324 -4.153 8.901 30.891   
  
Coefficients:  
 Estimate Std. Error t value Pr(>|t|)  
(Intercept) -1.411e+05 1.046e+04 -13.499 3.44e-08  
year 6.984e+01 5.166e+00 13.520 3.38e-08  
IndustryEducation -4.797e+01 1.461e+01 -3.283 0.0073  
IndustryFishing and Agriculture 2.803e+01 1.461e+01 1.918 0.0814  
IndustryGovernment Administration -1.131e+02 1.461e+01 -7.742 8.91e-06  
IndustryHealth and Community -1.402e+01 1.461e+01 -0.959 0.3581  
IndustryProperty and Business Service -3.382e+01 1.461e+01 -2.314 0.0410  
   
(Intercept) \*\*\*  
year \*\*\*  
IndustryEducation \*\*   
IndustryFishing and Agriculture .   
IndustryGovernment Administration \*\*\*  
IndustryHealth and Community   
IndustryProperty and Business Service \*   
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
  
Residual standard error: 17.89 on 11 degrees of freedom  
Multiple R-squared: 0.9638, Adjusted R-squared: 0.9441   
F-statistic: 48.84 on 6 and 11 DF, p-value: 2.7e-07









NULL

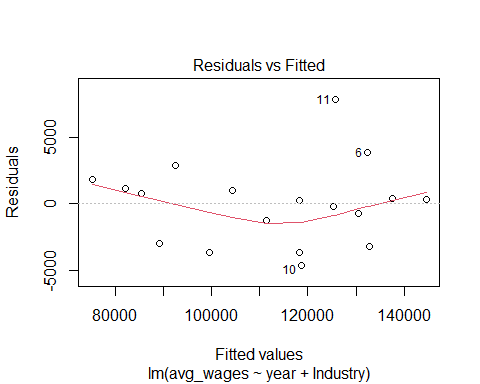
1   
98899.79

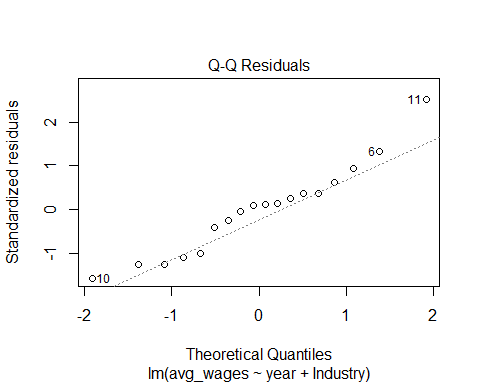
# Average wage per employee by industry ----  
wages\_table <- data %>%   
 select(`Company ID`, Industry, `2023` = `2023 wages`, `2024` = `2024 wages`, `2025` = `2025 wages`) %>%   
 pivot\_longer(cols = 3:5, names\_to = "year", values\_to = "wages") %>%   
 mutate(year = as.numeric(year)) %>%   
 group\_by(Industry, year) %>%   
 summarise(wages = sum(wages)) %>%   
 full\_join(employees\_table) %>%   
 group\_by(Industry, year) %>%   
 mutate(avg\_wages = wages / employees)

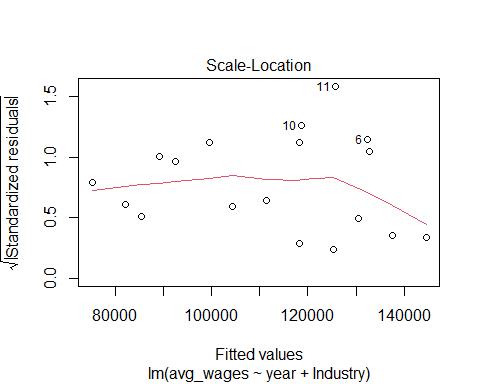
`summarise()` has grouped output by 'Industry'. You can override using the  
`.groups` argument.  
Joining with `by = join\_by(Industry, year)`

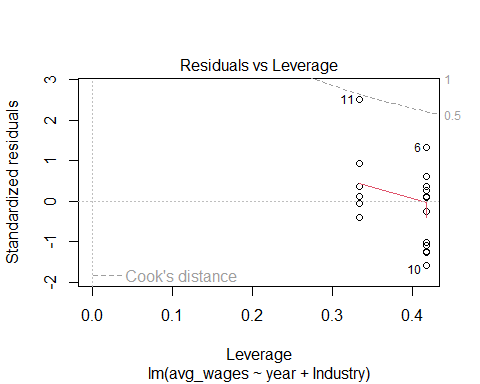
predict\_wage <- function(industry, summary = TRUE, assumptions = T) {  
 # input an industry and then output an average wage in 2026  
 model <- lm(avg\_wages ~ year + Industry, data = wages\_table)  
 if(summary) {print(summary(model))}  
 if(assumptions) {print(plot(model))}  
 predict(model, newdata = data.frame(year = 2026, Industry = industry)) %>%   
 return()  
}  
predict\_wage("Education")

Call:  
lm(formula = avg\_wages ~ year + Industry, data = wages\_table)  
  
Residuals:  
 Min 1Q Median 3Q Max   
-4619.8 -2548.1 286.4 1127.6 7838.2   
  
Coefficients:  
 Estimate Std. Error t value Pr(>|t|)   
(Intercept) -14087349 2235752 -6.301 5.83e-05 \*\*\*  
year 7001 1105 6.338 5.54e-05 \*\*\*  
IndustryEducation 43075 3124 13.787 2.76e-08 \*\*\*  
IndustryFishing and Agriculture 10317 3124 3.302 0.00705 \*\*   
IndustryGovernment Administration 43475 3124 13.915 2.51e-08 \*\*\*  
IndustryHealth and Community 29106 3124 9.316 1.49e-06 \*\*\*  
IndustryProperty and Business Service 55306 3124 17.702 1.97e-09 \*\*\*  
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
  
Residual standard error: 3827 on 11 degrees of freedom  
Multiple R-squared: 0.9788, Adjusted R-squared: 0.9672   
F-statistic: 84.58 on 6 and 11 DF, p-value: 1.472e-08









NULL

1   
139306.4

# Claim rate by industry ----  
claims\_table <- data %>%   
 select(`Company ID`, Industry, `2023` = `2023 number of claims`, `2024` = `2024 number of claims`, `2025` = `2025 number of claims`) %>%   
 pivot\_longer(cols = 3:5, names\_to = "year", values\_to = "claims") %>%   
 mutate(year = as.numeric(year)) %>%   
 group\_by(Industry, year) %>%   
 summarise(claims = sum(claims)) %>%   
 full\_join(employees\_table) %>%   
 group\_by(Industry, year) %>%   
 mutate(claim\_rate = sum(claims)/sum(employees))

`summarise()` has grouped output by 'Industry'. You can override using the  
`.groups` argument.  
Joining with `by = join\_by(Industry, year)`

predict\_claim\_rate <- function(industry) {  
 # input an industry and then output an average claim\_rate in 2026  
 # regression doesn't seem suitable but an average of the 3 years seems appropiate  
 claims\_table %>%   
 filter(Industry == industry) %>%   
 pull(claim\_rate) %>%   
 mean() %>%   
 return()  
}  
  
# NOT COMPLETE AND DODGY RN Average medical cost per claim by industry ----  
 # possibly using survival analysis technique? like KME or whatever  
costs\_table <- data %>%   
 select(Industry, `Company ID`, `2023` = `2023 medical costs`, `2024` = `2024 medical costs`, `2025` = `2025 medical costs`) %>%   
 pivot\_longer(cols = 3:5, names\_to = "year", values\_to = "costs") %>%  
 mutate(year = as.numeric(year)) %>%   
 group\_by(Industry, year) %>%   
 summarise(costs = sum(costs)) %>%   
 full\_join(claims\_table) %>%   
 group\_by(Industry, year) %>%   
 mutate(avg\_costs = costs / claims) %>%   
 select(Industry, year, costs, claims, avg\_costs)

`summarise()` has grouped output by 'Industry'. You can override using the  
`.groups` argument.  
Joining with `by = join\_by(Industry, year)`

predict\_costs <- function(industry) {  
 # input an industry and then output an average medical costs per claim in 2026  
 # THIS IS EXTREMELY PLACE HOLDER RN UNTIL WE ACTUALLY MODEL THE CLAIMS COSTS GOODLY  
 costs\_table %>%   
 filter(Industry == industry) %>%   
 pull(avg\_costs) %>%   
 mean() %>%   
 return()  
}  
  
# employees\_2026 <- c(  
# predict\_employees("Accommodation and Hospitality"),  
# predict\_employees("Government Administration"),  
# predict\_employees("Fishing and Agriculture"),  
# predict\_employees("Education"),  
# predict\_employees("Health and Community"),  
# predict\_employees("Property and Business Service")  
# )  
#   
# avg\_wages\_2026 <- c(  
# predict\_wage("Accommodation and Hospitality"),  
# predict\_wage("Government Administration"),  
# predict\_wage("Fishing and Agriculture"),  
# predict\_wage("Education"),  
# predict\_wage("Health and Community"),  
# predict\_wage("Property and Business Service")  
# )  
#   
# claim\_rate\_2026 <- c(  
# predict\_claim\_rate("Accommodation and Hospitality"),  
# predict\_claim\_rate("Government Administration"),  
# predict\_claim\_rate("Fishing and Agriculture"),  
# predict\_claim\_rate("Education"),  
# predict\_claim\_rate("Health and Community"),  
# predict\_claim\_rate("Property and Business Service")   
# )  
#   
# avg\_costs\_2026 <- c(  
# predict\_costs("Accommodation and Hospitality"),  
# predict\_costs("Government Administration"),  
# predict\_costs("Fishing and Agriculture"),  
# predict\_costs("Education"),  
# predict\_costs("Health and Community"),  
# predict\_costs("Property and Business Service")  
# )  
  
# predicted\_data <- employees\_table %>%   
# full\_join(wages\_table) %>%   
# full\_join(claims\_table) %>%   
# full\_join(costs\_table) %>%   
# select(Industry, year, employees, avg\_wages, claim\_rate, avg\_costs) %>%   
# ungroup() %>%   
# add\_row(Industry = c("Accommodation and Hospitality",  
# "Government Administration",  
# "Fishing and Agriculture",  
# "Education",  
# "Health and Community",  
# "Property and Business Service"),  
# year = 2026,  
# employees = employees\_2026,  
# avg\_wages = avg\_wages\_2026,  
# claim\_rate = claim\_rate\_2026,  
# avg\_costs = avg\_costs\_2026)

## Predicting the factors making up the costs

* Modelling was done using either linear regression or the average over the past 3 years
  + Claim\_rate appears constant for each industry so the average was used
  + Sqrt(employees) followed a linear trend
  + Average wages followed a linear trend
  + Average medical cost per claim was averaged REVISE THIS IF WE MAKE A BETTER METHOD LATER

# plot0 <- predicted\_data %>%   
# ggplot(aes(x=year, y=employees, colour = Industry)) +  
# geom\_line() +  
# geom\_point() +   
# labs(  
# title = "Employees over time by industry",  
# x = "Year",  
# y = "Employees"   
# ) +  
# theme\_light()  
#   
# plot1 <- predicted\_data %>%   
# ggplot(aes(x=year, y=avg\_wages, colour = Industry)) +  
# geom\_line() +  
# geom\_point() +   
# labs(  
# title = "avg\_wages over time by industry",  
# x = "Year",  
# y = "avg\_wages"   
# ) +  
# theme\_light()  
#   
# plot2 <- predicted\_data %>%   
# ggplot(aes(x=year, y=claim\_rate, colour = Industry)) +  
# geom\_line() +  
# geom\_point() +   
# labs(  
# title = "claim\_rate over time by industry",  
# x = "Year",  
# y = "claim\_rate"   
# ) +  
# theme\_light()  
#   
# plot3 <- predicted\_data %>%   
# ggplot(aes(x=year, y=avg\_costs, colour = Industry)) +  
# geom\_line() +  
# geom\_point() +   
# labs(  
# title = "avg\_costs over time by industry",  
# subtitle = "cant fully justify the modelling yet",  
# x = "Year",  
# y = "avg\_costs"   
# ) +  
# theme\_light()  
#   
# plot0 + plot1 + plot2 + plot3 + plot\_layout(guides = "collect") & theme\_light(base\_size = 14) & theme(legend.position = "bottom")

## Combinging it to get our numbers

# total\_medical\_costs <- (claim\_rate\_2026 \* employees\_2026 \* avg\_costs\_2026) %>% sum()  
# total\_salary\_compensation <- (0.7 \* 4/52 \* claim\_rate\_2026 \* employees\_2026 \* avg\_wages\_2026) %>% sum()  
# total\_costs <- total\_medical\_costs + total\_salary\_compensation + 2400000