lab3\_luzanova\_ep61

# Загрузимо ланні та Видалемо всі незначущі змінні, залишимо Population, Life\_expect, Empl\_serv.

f\_train <- read.csv2('crime\_train.csv', header = TRUE, encoding = 'UNICOD')  
f\_train <- f\_train[,-c(1:2,5:7,9:10,12:18)]  
f\_test <- read.csv2('crime\_test.csv', header = TRUE, encoding = 'UNICOD')  
f\_test <- f\_test[,-c(1:2, 5:7,9:10,12:18)]  
  
head(f\_train)

## Homicide Population Life\_expect Empl\_serv  
## 1 7.810353 17.73195 4.189655 3.626919  
## 2 3.496508 14.05028 4.317488 4.219508  
## 3 8.462737 17.53304 4.143135 3.626919  
## 4 4.867534 13.93307 4.130230 4.143135  
## 5 9.921082 17.85857 4.158883 4.262680  
## 6 5.480639 16.77007 4.130230 3.688879

head(f\_test)

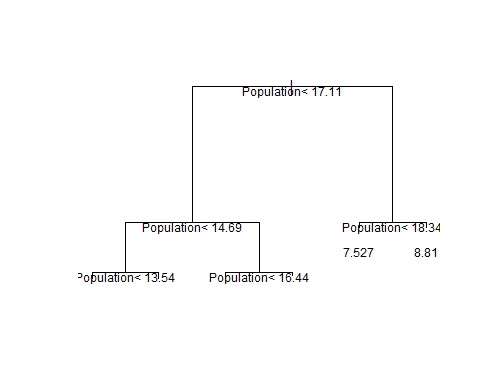
## Homicide Population Life\_expect Empl\_serv  
## 1 5.831882 17.01688 4.130230 3.688879  
## 2 6.634633 17.38733 4.330733 3.737670  
## 3 4.836282 16.22921 4.130230 3.713572  
## 4 6.411818 17.18699 4.143135 3.871201  
## 5 7.103322 16.16814 4.304065 4.262680  
## 6 3.970292 12.67382 4.330733 4.158883

#Висновок: окремо завантажені навчальна і тестова вибірки. # Decision Tree Regression ## Fitting

library(rpart)

## Warning: package 'rpart' was built under R version 3.6.3

model\_dt <- rpart(Homicide ~ Population, f\_train, control = rpart.control(minsplit = 10))  
plot(model\_dt)  
text(model\_dt, pos = 1, cex = 0.75, col = 1, font = 1)

 #Висновок: побудовано дерево рішень, екзогенна змінна – Population.

## Predicting

p\_dt <- predict(model\_dt, f\_test)  
  
train\_mse\_dt <- sum((f\_train$Homicide-predict(model\_dt, f\_train))^2)/length(f\_train$Homicide)  
test\_mse\_dt <- sum((f\_test$Homicide-p\_dt)^2)/length(p\_dt)  
  
r2\_dt <- 1-sum((f\_train$Homicide - predict(model\_dt, f\_train))^2)/sum((f\_train$Homicide - mean(f\_train$Homicide))^2)  
  
r2\_dt

## [1] 0.7197324

train\_mse\_dt

## [1] 1.510216

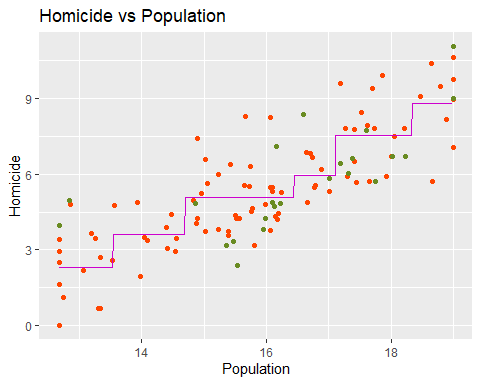
test\_mse\_dt

## [1] 2.187112

#Висновок: значення середньоквадратичної похибки покращилися на навчальній вибірці – 1.51, погіршилися на тестовій вибірці – 2.187. Модель перенавчено.

## Visualising

library(ggplot2)  
x\_grid <- seq(min(f\_train$Population), max(f\_train$Population), 0.01)  
ggplot() +  
 geom\_point(aes(f\_train$Population, f\_train$Homicide),colour = 'orangered') +  
 geom\_point(aes(f\_test$Population, f\_test$Homicide),colour = 'olivedrab4') +  
 geom\_line(aes(x\_grid, predict(model\_dt, data.frame(Population = x\_grid))),colour = 'magenta3') +  
 ggtitle('Homicide vs Population') +  
 xlab('Population') +  
 ylab('Homicide')

 #Висновок: на графіку червоним позначені точки навчальної вибірки, зеленим – точки тестової вибірки, фіолетовим – модельні значення.

# Random forest

## Fitting

library(randomForest)

## Warning: package 'randomForest' was built under R version 3.6.3

## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##   
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':  
##   
## margin

set.seed(1234)  
model\_rf = randomForest(x = f\_train['Population'],  
 y = f\_train$Homicide,  
 ntree = 50)

#Висновок: побудовано віпадковий ліс із 50 дерев, екзогенна змінна – Population.

## Predicting

p\_rf <- predict(model\_rf, f\_test)  
  
train\_mse\_rf <- sum((f\_train$Homicide-predict(model\_rf, f\_train))^2)/length(f\_train$Homicide)  
test\_mse\_rf <- sum((f\_test$Homicide-p\_rf)^2)/length(p\_rf)  
r2\_rf <- 1-sum((f\_train$Homicide - predict(model\_rf, f\_train))^2)/sum((f\_train$Homicide - mean(f\_train$Homicide))^2)  
  
r2\_rf

## [1] 0.8672666

train\_mse\_rf

## [1] 0.7152313

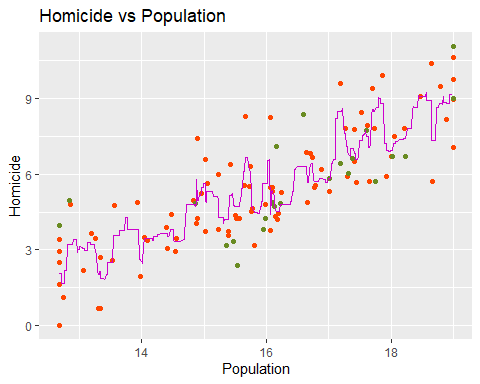
test\_mse\_rf

## [1] 2.152029

#Висновок: значення середньоквадратичної похибки покращилися на навчальній вибірці – 0.715, покращилися на тестовій вибірці – 2,152. Модель перенавчено.

## Visualising

ggplot() +  
 geom\_point(aes(f\_train$Population, f\_train$Homicide),colour = 'orangered') +  
 geom\_point(aes(f\_test$Population, f\_test$Homicide),colour = 'olivedrab4') +  
 geom\_line(aes(x\_grid, predict(model\_rf, data.frame(Population = x\_grid))),colour = 'magenta3') +  
 ggtitle('Homicide vs Population') +  
 xlab('Population') +  
 ylab('Homicide')

 #Висновок: на графіку червоним позначені точки навчальної вибірки, зеленим – точки тестової вибірки, фіолетовим – модельні значення.

#перейдемо до Нейронних мереж

library(dplyr)

## Warning: package 'dplyr' was built under R version 3.6.3

##   
## Attaching package: 'dplyr'

## The following object is masked from 'package:randomForest':  
##   
## combine

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(nnet)

## Warning: package 'nnet' was built under R version 3.6.3

library(ggplot2)  
library(knitr)  
library (psych)

## Warning: package 'psych' was built under R version 3.6.3

##   
## Attaching package: 'psych'

## The following object is masked from 'package:randomForest':  
##   
## outlier

## The following objects are masked from 'package:ggplot2':  
##   
## %+%, alpha

#проведемо шкалювання окремо на тестовій виборці, окремо на тренувальній #```{r} #f\_sc\_train <- f\_train #f\_sc\_trainHomicide) #f\_sc\_trainPopulation) #f\_sc\_trainLife\_expect) #f\_sc\_trainEmpl\_serv) #f\_train<-f\_sc\_train

#f\_sc\_test <- f\_test #f\_sc\_testHomicide) #f\_sc\_testPopulation) #f\_sc\_testLife\_expect) #f\_sc\_testEmpl\_serv) #f\_test <- f\_sc\_test

#head (f\_train) #head (f\_test) #```

set.seed(333)  
ff\_ap <- nnet(data = f\_train, Homicide ~ Population + Life\_expect + Empl\_serv, linout = TRUE ,size = 12, maxit = 10000)

## # weights: 61  
## initial value 1618.430605   
## iter 10 value 197.134367  
## iter 20 value 128.521031  
## iter 30 value 110.532625  
## iter 40 value 107.447200  
## iter 50 value 106.757770  
## iter 60 value 104.089634  
## iter 70 value 94.814451  
## iter 80 value 78.140624  
## iter 90 value 71.712328  
## iter 100 value 71.307741  
## iter 110 value 70.926511  
## iter 120 value 70.832197  
## iter 130 value 70.349329  
## iter 140 value 69.468954  
## iter 150 value 68.394521  
## iter 160 value 66.288847  
## iter 170 value 65.674808  
## iter 180 value 65.530171  
## iter 190 value 65.312498  
## iter 200 value 65.294493  
## iter 210 value 65.274901  
## iter 220 value 65.202577  
## iter 230 value 64.840178  
## iter 240 value 63.500046  
## iter 250 value 62.853333  
## iter 260 value 62.499065  
## iter 270 value 62.465347  
## iter 280 value 62.432622  
## iter 290 value 62.334093  
## iter 300 value 62.289482  
## iter 310 value 62.280938  
## iter 320 value 62.263002  
## iter 330 value 62.257758  
## iter 340 value 62.245571  
## iter 350 value 62.245516  
## iter 360 value 62.245191  
## iter 370 value 62.243965  
## iter 380 value 62.242848  
## iter 390 value 62.241267  
## iter 400 value 62.237362  
## iter 410 value 62.234646  
## iter 420 value 62.232381  
## final value 62.231048   
## converged

library(graphics)  
source(file = 'plot.nnet.R')  
plot.nnet(ff\_ap)

## Loading required package: scales

##   
## Attaching package: 'scales'

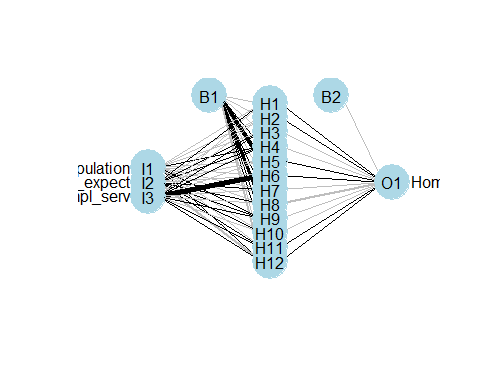
## The following objects are masked from 'package:psych':  
##   
## alpha, rescale

## Loading required package: reshape

## Warning: package 'reshape' was built under R version 3.6.3

##   
## Attaching package: 'reshape'

## The following object is masked from 'package:dplyr':  
##   
## rename

 #Висновок: на основі змінних Population, Life\_expect, Empl\_serv побудовано двошарову нейронну мережу для прогнозування Homicide

p\_ff\_ap <- predict(ff\_ap, f\_test)  
  
train\_mse\_ff\_ap <- sum((f\_train$Homicide-predict(ff\_ap, f\_train))^2)/length(f\_train$Homicide)  
test\_mse\_ff\_ap <- sum((f\_test$Homicide-p\_ff\_ap)^2)/length(p\_ff\_ap)  
  
r2\_ff\_ap <- 1-sum((f\_train$Homicide - predict(ff\_ap, f\_train))^2)/sum((f\_train$Homicide - mean(f\_train$Homicide))^2)  
  
r2\_ff\_ap

## [1] 0.8716788

train\_mse\_ff\_ap

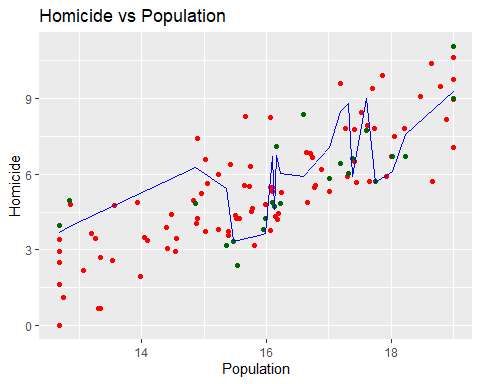
## [1] 0.6914561

test\_mse\_ff\_ap

## [1] 1.767834

#Висновок: значення середньоквадратичної похибки на навчальній вибірці – 0.032, на тестовій вибірці – 0.65. Модель перенавчено.

library(ggplot2)  
ggplot() +  
 geom\_point(aes(f\_train$Population, f\_train$Homicide),colour = 'red') +  
 geom\_point(aes(f\_test$Population, f\_test$Homicide),colour = 'darkgreen') +  
 geom\_line(aes(f\_test$Population, p\_ff\_ap),colour = 'blue') +  
 ggtitle('Homicide vs Population') +  
 xlab('Population') +  
 ylab('Homicide')

 #Висновок: на графіку червоним позначені точки навчальної вибірки, зеленим – точки тестової вибірки, фіолетовим – модельні значення.

fit <- read.csv2('crime\_fit.csv', header = TRUE, encoding = 'UNICOD')  
fit <- fit[,-c(1)]  
fit$f\_test\_Homicide <-f\_test$Homicide  
fit$p\_dt <- p\_dt  
fit$p\_rf <- p\_rf  
fit$p\_ff\_ap <- p\_ff\_ap  
head(fit)

## p\_sr p\_mr p\_pr f\_test\_Homicide p\_dt p\_rf p\_ff\_ap  
## 1 6.646040 8.159338 6.591823 5.831882 5.970051 5.850473 7.079924  
## 2 7.041017 6.275327 7.019894 6.634633 7.526866 7.027045 5.931860  
## 3 5.806235 7.353415 5.710147 4.836282 5.078873 4.792396 6.046321  
## 4 6.827411 8.553121 6.787329 6.411818 7.526866 8.473266 8.484163  
## 5 5.741120 6.312725 5.643411 7.103322 5.078873 4.517080 6.754971  
## 6 2.015502 1.996263 2.219673 3.970292 2.305050 2.058364 3.690355

write.csv2(fit, file = "crime\_fit.csv")

test\_mse <- read.csv2('crime\_test\_mse.csv', header = TRUE, encoding = 'UNICOD')  
test\_mse <- test\_mse[,-c(1)]  
test\_mse$test\_mse\_dt <- test\_mse\_dt  
test\_mse$test\_mse\_rf <- test\_mse\_rf  
test\_mse$test\_mse\_ff\_ap <- test\_mse\_ff\_ap  
head(test\_mse)

## test\_mse\_sr test\_mse\_opt test\_mse\_poly test\_mse\_dt test\_mse\_rf test\_mse\_ff\_ap  
## 1 2.231931 2.227331 2.056081 2.187112 2.152029 1.767834

write.csv2(test\_mse, file = "crime\_test\_mse.csv")  
  
train\_mse <- read.csv2('crime\_train\_mse.csv', header = TRUE, encoding = 'UNICOD')  
train\_mse <- train\_mse[,-c(1)]  
train\_mse$train\_mse\_dt <- train\_mse\_dt  
train\_mse$train\_mse\_rf <- train\_mse\_rf  
train\_mse$train\_mse\_ff\_ap <- train\_mse\_ff\_ap  
head(train\_mse)

## train\_mse\_sr train\_mse\_opt train\_mse\_poly train\_mse\_dt train\_mse\_rf  
## 1 1.696169 1.204248 1.685496 1.510216 0.7152313  
## train\_mse\_ff\_ap  
## 1 0.6914561

write.csv2(train\_mse, file = "crime\_train\_mse.csv")  
  
r2 <- read.csv2('crime\_r2.csv', header = TRUE, encoding = 'UNICOD')  
r2 <- r2[,-c(1)]  
r2$r2\_dt <- r2\_dt  
r2$r2\_rf <- r2\_rf  
r2$r2\_ff\_ap <- r2\_ff\_ap  
head(r2)

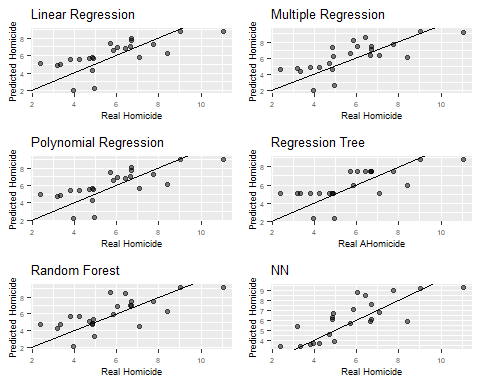
## r2\_sr r2\_opt R2\_pr r2\_dt r2\_rf r2\_ff\_ap  
## 1 0.685223 0.7765142 0.6872038 0.7197324 0.8672666 0.8716788

write.csv2(r2, file = "crime\_r2.csv")

f1 <- read.csv2('crime\_test.csv.', header = TRUE, encoding = 'UNICOD')  
f2 <- read.csv2('crime\_fit.csv', header = TRUE, encoding = 'UNICOD')  
f2 <- f2[-1]  
f <- dplyr::bind\_cols(f1, f2)  
f <- f[,-1]  
f$Population <- as.factor(f$Population)  
head(f)

## Territory Homicide Population GDP Unempl  
## 1 Cameroon 5.831882 17.0168757628795 24.27888 1.0986123  
## 2 Morocco 6.634633 17.3873306082871 25.42375 2.1972246  
## 3 Benin 4.836282 16.2292079533308 23.26500 0.6931472  
## 4 Ghana 6.411818 17.1869862951268 24.80077 1.3862944  
## 5 Dominican Republic 7.103322 16.1681356052432 25.10527 1.7917595  
## 6 Saint Lucia 3.970292 12.6738153138436 21.41596 2.9520320  
## Agri\_fore\_fish\_GDP Life\_expect Death\_rate Birth\_rate Empl\_serv Empl\_agri  
## 1 22.33901 4.130230 2.302585 3.583519 3.688879 3.806662  
## 2 23.33036 4.330733 1.609438 2.944439 3.737670 3.583519  
## 3 22.00935 4.130230 2.197225 3.610918 3.713572 3.688879  
## 4 23.17602 4.143135 1.945910 3.401197 3.871201 3.465736  
## 5 22.17582 4.304065 1.791759 2.995732 4.262680 2.302585  
## 6 18.43200 4.330733 1.945910 2.484907 4.158883 2.890372  
## Empl\_indu educ\_expen Mil\_expen Arm\_forc\_person Region Subregion  
## 1 2.708050 20.64045 19.82716 10.085809 Africa Middle Africa  
## 2 3.091042 22.44484 21.96496 12.413087 Africa Northern Africa  
## 3 2.944439 20.04178 18.57033 9.392662 Africa Western Africa  
## 4 2.995732 21.61426 19.05794 9.680344 Africa Western Africa  
## 5 2.944439 21.37547 20.09418 11.170435 Americas Caribbean  
## 6 2.890372 18.10739 21.00000 11.000000 Americas Caribbean  
## p\_sr p\_mr p\_pr f\_test\_Homicide p\_dt p\_rf p\_ff\_ap  
## 1 6.646040 8.159338 6.591823 5.831882 5.970051 5.850473 7.079924  
## 2 7.041017 6.275327 7.019894 6.634633 7.526866 7.027045 5.931860  
## 3 5.806235 7.353415 5.710147 4.836282 5.078873 4.792396 6.046321  
## 4 6.827411 8.553121 6.787329 6.411818 7.526866 8.473266 8.484163  
## 5 5.741120 6.312725 5.643411 7.103322 5.078873 4.517080 6.754971  
## 6 2.015502 1.996263 2.219673 3.970292 2.305050 2.058364 3.690355

g\_sr <- ggplot(f, aes(x=Homicide, y=p\_sr)) +   
 geom\_abline(intercept=0, slope=1) +  
 geom\_point( alpha=0.5) + labs(title="Linear Regression", x="Real Homicide", y="Predicted Homicide") +   
 theme(plot.title=element\_text(size=10), axis.title.x=element\_text(size=7), axis.title.y=element\_text(size=7), axis.text.x=element\_text(size=5), axis.text.y=element\_text(size=5)) + theme(legend.position="none")  
  
g\_mr <- ggplot(f, aes(x=Homicide, y=p\_mr)) +   
 geom\_abline(intercept=0, slope=1) +  
 geom\_point(alpha=0.5) + labs(title="Multiple Regression", x="Real Homicide", y="Predicted Homicide") +   
 theme(plot.title=element\_text(size=10), axis.title.x=element\_text(size=7), axis.title.y=element\_text(size=7), axis.text.x=element\_text(size=5), axis.text.y=element\_text(size=5)) + theme(legend.position="none")  
  
g\_pr <- ggplot(f, aes(x=Homicide, y=p\_pr)) +   
 geom\_abline(intercept=0, slope=1) +  
 geom\_point(alpha=0.5) + labs(title="Polynomial Regression", x="Real Homicide", y="Predicted Homicide") +   
 theme(plot.title=element\_text(size=10), axis.title.x=element\_text(size=7), axis.title.y=element\_text(size=7), axis.text.x=element\_text(size=5), axis.text.y=element\_text(size=5)) + theme(legend.position="none")   
  
g\_dt <- ggplot(f, aes(x=Homicide, y=p\_dt)) +   
 geom\_abline(intercept=0, slope=1) +  
 geom\_point(alpha=0.5) + labs(title="Regression Tree", x="Real AHomicide", y="Predicted Homicide") +   
 theme(plot.title=element\_text(size=10), axis.title.x=element\_text(size=7), axis.title.y=element\_text(size=7), axis.text.x=element\_text(size=5), axis.text.y=element\_text(size=5)) + theme(legend.position="none")  
  
g\_rf <- ggplot(f, aes(x=Homicide, y=p\_rf)) +   
 geom\_abline(intercept=0, slope=1) +  
 geom\_point( alpha=0.5) + labs(title="Random Forest", x="Real Homicide", y="Predicted Homicide") +   
 theme(plot.title=element\_text(size=10), axis.title.x=element\_text(size=7), axis.title.y=element\_text(size=7), axis.text.x=element\_text(size=5), axis.text.y=element\_text(size=5)) + theme(legend.position="none")  
  
g\_ff\_ap <- ggplot(f, aes(x=Homicide, y=p\_ff\_ap)) +   
 geom\_abline(intercept=0, slope=1) +  
 geom\_point( alpha=0.5) + labs(title="NN", x="Real Homicide", y="Predicted Homicide") +   
 theme(plot.title=element\_text(size=10), axis.title.x=element\_text(size=7), axis.title.y=element\_text(size=7), axis.text.x=element\_text(size=5), axis.text.y=element\_text(size=5)) + theme(legend.position="none")  
  
gridExtra::grid.arrange(g\_sr,g\_mr,g\_pr,g\_dt,g\_rf,g\_ff\_ap,ncol=2)



# Calc prediction error and visualize it (хоча ми це зробили, але іншим шляхом)

sr <- mean ((f$Homicide - f$p\_sr) ^ 2)  
mr <- mean ((f$Homicide - f$p\_mr) ^ 2)  
pr <- mean ((f$Homicide - f$p\_pr) ^ 2)  
dt <- mean ((f$Homicide - f$p\_dt) ^ 2)  
rf <- mean ((f$Homicide - f$p\_rf) ^ 2)  
ff\_ap <- mean ((f$Homicide - f$p\_ff\_ap) ^ 2)  
mse <- data.frame(sr,mr,pr,dt,rf,ff\_ap)  
head(mse)

## sr mr pr dt rf ff\_ap  
## 1 2.231931 2.227331 2.056081 2.187112 2.152029 1.767834

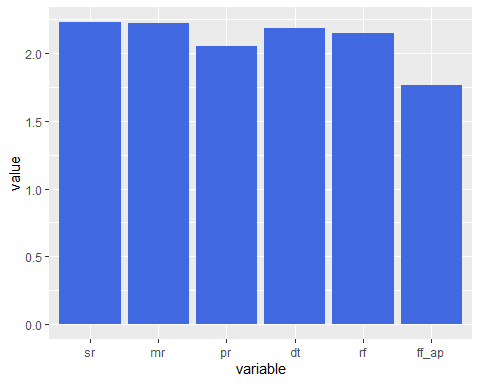
library(reshape)  
mse1 <- reshape::melt.data.frame(mse)

## Using as id variables

head(mse1)

## variable value  
## 1 sr 2.231931  
## 2 mr 2.227331  
## 3 pr 2.056081  
## 4 dt 2.187112  
## 5 rf 2.152029  
## 6 ff\_ap 1.767834

b1 <- ggplot(mse1, aes(x=variable, y=value)) +  
 geom\_bar(stat="summary", fun.y="mean", fill = 'royalblue')  
b1



ggsave("plot.jpg", plot=b1 + theme\_classic(), width=20, height=15, units="cm", dpi=600)

#Висновок: побудовано моделі лінійної регресії, багатофакторної регресії, поліноміальної регресії, дерево рішень, випадковий ліс, нейронної мережі. Розроблено прогноз по кожній з моделей та проведено аналіз на базі тестувальних даних. Зазначимо, що всі моделі мають явище перенавчання. Найменше значення середньоквадратичної похибки на навчальній виборці та тестовій виборці - у моделі нейронної мережі

#Явище перенавчання присутне за причиною замалої кількості спостережень. Найбільше значення коефіцієнту детермінації у моделі нейронної мережі

#Аналізуючи значення коефіцієнтів детермінації робимо висновок, що модель нейронної мережі найбільш точно описує зв’зок та дає найбільш точний прогноз. Це дає підстави зробити висновки про найвищу точність прогнозу з усіх розглянутих моделей