8i9

library(ggplot2)

library(PogromcyDanych)

setLang(lang = 'eng')

library(dplyr) #przeksztalcanie danych

library(tidyr)

#1 ####

cats\_birds %>%

ggplot(aes(x = length, y = speed))+

geom\_point(aes(color = group, shape = group), size = 3)+

geom\_smooth(method = "lm", se = FALSE, color = 'black')+

theme\_minimal()

#2 ####

pearson %>%

ggplot(aes(x = father, y = son))+

geom\_point()+

geom\_smooth(method = "lm", se = FALSE, color = 'green')+

theme\_minimal()

#3 ####

set.seed(100)

sample(unique(seriesIMDB$series), size = 20, replace = FALSE) -> selected.series

seriesIMDB %>%

filter(series %in% selected.series) %>%

mutate(series = reorder(series, note, median)) %>% #posortowanie po mednianie ocen,

ggplot(aes(x = series, y = note))+

geom\_boxplot()+

coord\_flip() #zamiana osi

#4 ####

skimr::skim(diagnosis)

levels(diagnosis$gp29)

diagnosis%>%

select(eduk4\_2013, gp29) %>%

na.omit() %>%

ggplot(aes(x = eduk4\_2013, fill = gp29))+

geom\_bar()+

coord\_flip()

#5 #####

library(ggthemes)

auta2012 %>%

filter(Brand == 'Volkswagen'& Model == 'Passat') %>%

ggplot(aes(x = Year, y = Price.in.PLN))+

geom\_point()+

geom\_smooth(se = FALSE)+

theme\_gdocs()

#6 ####

ggplot(cats\_birds, aes(x = weight, y = speed, size = lifespan, color = lifespan)) +

geom\_point(shape = 15)+

scale\_color\_gradient(low = 'green', high = 'red')+

ggtitle('TytuĹ‚')+

xlab("Waga [kg]")+

ylab('PrÄ™dkoĹ›Ä‡ [km/h]')

#7 ####

auta2012 %>%

filter(Brand == 'Toyota') %>%

count(Model) %>%

slice\_max(n, n=5) %>%

mutate(Model = reorder(Model, n)) %>%

ggplot(aes(x = '', y = n, fill = Model))+

geom\_col()+

coord\_polar('y')+

guides(fill = 'none')+

theme\_minimal()

#8 #####

pearson %>%

mutate(id=1:nrow(pearson)) %>%

pivot\_longer(cols=c('son','father'),

values\_to='Height',

names\_to='who') %>%

ggplot(aes(x=Height)) +

geom\_histogram(aes(y=..density..))+

geom\_density(color='purple',size=2)+

facet\_wrap('who',ncol=2)

#9 #####

iris %>%

ggplot(aes(x=Sepal.Length ,y=Sepal.Width,shape=Species,col=Species)) +

geom\_density2d(n=100,h=c(0.8,0.8))+

geom\_point()+

theme\_light()

#10 ####

iris %>%

bind\_rows(iris %>% mutate(Species="ALL")) %>% #dokleja kopie iris do iris

ggplot(aes(x=Petal.Length,y=Petal.Width,col=Species))+

geom\_point()+

geom\_smooth()+

guides(color='none')+

facet\_wrap('Species',nrow=2,ncol=2,scales='free') #rozdziela na 4 wykresy (Ćwiartki)

#11 ####

mtcars %>%

as\_tibble(rownames='Model') %>%

mutate(Model=reorder(Model,mpg)) %>%

ggplot(aes(x=mpg,y=Model,label=Model))+

geom\_point() +

geom\_text(hjust=0,nudge\_x=0.4)+

theme\_minimal()+

xlim(c(10,40))+

theme(axis.title.y = element\_blank(),

axis.ticks.y = element\_blank(),

axis.line.y = element\_blank(),

axis.text.y = element\_blank())

#12 ####

library(ggthemes)

economics %>%

ggplot(aes(x = date,

y = uempmed))+

geom\_line() +

theme\_economist\_white()+

labs(x= '',

y= 'Median duration of unemployment (weeks)',

title= 'Median duration of unemployment (weeks)')

z pakietu ggthemes.

10

#1 ####

library(tidyr)

library(lubridate)

#a

start.date <- dmy('23012019')

class(start.date)

as.numeric(start.date) #ile dni minelo od 1.1.1970

#b

today()

#c

year(start.date)

#d

month(start.date,label=TRUE,abbr=FALSE)

#e

day(start.date)

#f

wday(start.date,label=TRUE,abbr=FALSE)

#g

month(start.date) <- 2

#h

start.date + days(6)

#i

start.date - months(3)

#j

start.date + 1:10

#k

end.date <- dmy\_hms('01/10/2019 16:01:10')

#l

grep('Buenos',OlsonNames(),value=TRUE)

with\_tz(end.date, tzone='America/Argentina/Buenos\_Aires')

#2 ####

library(dplyr)

library(tidyr)

library(stringr)

library(gapminder)

library(ggplot2)

#a

gapminder %>%

select(country,continent) %>%

distinct ->df

View(df)

#b

df %>%

mutate(c.length = str\_length(country)) %>%

summarise(mean.l = mean(c.length))

#c

df %>%

mutate(pierwsza = str\_sub(country,1,1),

ostatnia = str\_sub(country,-1,-1)) %>%

select(pierwsza,ostatnia) %>%

pivot\_longer(everything()) %>%

ggplot(aes(x=value)) +

geom\_bar()+

facet\_grid(col=vars(name),

scales = 'free') +

theme\_minimal()

#d

df %>%

filter(str\_detect(country,' and '))

#e

df %>%

mutate(country=str\_remove\_all(country,'[\\.,]'))

#f

df %>%

mutate(skr=str\_trunc(as.vector(country),12,ellipsis='.'))->skr

View(skr)

#g

str\_to\_lower(df$country) %>%

str\_remove\_all('[\\.,]') %>%

str\_trim() %>%

str\_c(sep = "", collapse ='') %>%

str\_split('') %>%

unlist() %>%

as\_tibble() %>%

ggplot(aes(x=value))+

geom\_bar()+

theme\_minimal()

#h

df %>%

mutate(ilea={str\_to\_lower(country) %>%

str\_count('a')}) %>%

arrange(-ilea)

#3 ####

library(purrr)

#a)

map\_dbl(mtcars, mean)

#b)

plot.function<-function(df){

ggplot(df,aes(x = hp, y = qsec)) +

geom\_point()+

geom\_smooth(method='lm',

se = FALSE)

}

mtcars%>%

group\_by(cyl)%>%

tidyr::nest() %>%

mutate(n = map\_dbl(.x = data,

.f=function(x) nrow(x)))%>%

mutate(model.lm=map(.x =data,

.f=function(x) lm(qsec ~ hp,

data = x)))%>%

mutate(tab.coef=map(.x = model.lm,

.f=function(x) coef(x)))%>%

mutate(summary.lm=map(.x = model.lm,

.f=function(x) summary(x)))%>%

mutate(plot.lm = map(.x = data,

.f = function(x) plot.function(x))) -> df.final

12

library(shiny)

library(dplyr)

#1 ####

ui <- fluidPage(

sliderInput("x", label = "If x is", min = 1, max = 50, value = 30),

sliderInput("y", label = "and y is", min=1, max = 50, value = 5),

"then x times 5 is",

textOutput("product")

)

server <- function(input, output, session) {

output$product <- renderText({

input$x \* input$y

})

}

shinyApp(ui, server)

#3 ####

ui <- fluidPage(

sliderInput("x", "If x is", min = 1, max = 50, value = 30),

sliderInput("y", "and y is", min = 1, max = 50, value = 5),

"then, (x \* y) is", textOutput("product"),

"and, (x \* y) + 5 is", textOutput("product\_plus5"),

"and (x \* y) + 10 is", textOutput("product\_plus10")

)

server <- function(input, output, session) {

productXY <- reactive({

input$x \* input$y

})

output$product <- renderText({

product.value <- productXY()

product.value

})

output$product\_plus5 <- renderText({

product.value <- productXY()

product.value + 5

})

output$product\_plus10 <- renderText({

product.value <- productXY()

product.value + 10

})

}

shinyApp(ui, server)

#4 ####

library(dplyr)

library(purrr)

library(ggplot2)

library(highcharter)

#cwicz 11

#1 ####

library(plotly)

LakeHuron %>%

as\_tibble() %>%

mutate(Time = time(LakeHuron)) %>%

plot\_ly(x = ~Time, y = ~x) %>%

add\_lines()

#2 ####

iris %>%

plot\_ly(x = ~Petal.Length,

y = ~Petal.Width,

text = ~Species) %>%

add\_markers()

#3 ####

iris %>%

plot\_ly(x = ~Petal.Length,

y = ~Petal.Width,

color = ~Species) %>%

add\_markers()

#4 ####

esoph %>%

count(agegp)%>%

plot\_ly(x = ~agegp, y= ~n) %>%

add\_bars()

#5 ####

volcano %>%

plot\_ly(z = .,

type = "heatmap")

#6 ####

devtools::install\_github("tutuchan/morrisjs") # Instalacja

library(morrisjs)

morrisjs(list(labels= letters[1:6],

values = c(0.12, 0.3, 0.26,

0.16, 0.04, 0.12))) %>%

mjsDonut(options(colors = c('#66c2ff', '#ff66ff',

'#80ffd4', '#b3b3cc',

'#ffff80')))

#7 ####

iris %>%

plot\_ly(x = ~Petal.Length) %>%

add\_histogram() -> p1

iris %>%

plot\_ly(x = ~Petal.Width) %>%

add\_histogram() -> p2

iris %>%

plot\_ly(x = ~Sepal.Length) %>%

add\_histogram() -> p3

iris %>%

plot\_ly(x = ~Petal.Width) %>%

add\_histogram() -> p4

suplot(p1,p2,p3,p4,

nrows=2)