**SUPERMARKET MANAGEMENT SYSTEM**



**CODING:**

# Install and load packages

library(tidyverse)

library(readr)

library(ggplot2)

library(ggalt)

# Load and sweep data

analyst\_jobs\_raw <-

read\_csv("~/Rstudio WD/DataAnalyst.csv") %>%

janitor::clean\_names()

View(analyst\_jobs\_raw)

# Get a quick look at our data using head() and skim()

skimr::skim(analyst\_jobs\_raw)

head(analyst\_jobs\_raw)

# Data Cleaning

analyst\_df <-

analyst\_jobs\_raw %>%

transmute(

company\_name = gsub("[[:digit:]]","",

gsub("\\.","",

company\_name)),

job\_title = str\_extract(job\_title,

pattern = "^([^,])+"),

job\_description,

location,

rating = case\_when(rating != -1 ~ as.numeric(rating), TRUE ~ NA\_real\_),

founded = case\_when(founded != -1 ~ as.numeric(founded), TRUE ~ NA\_real\_),

industry = case\_when(industry != "-1" ~ as.character(industry), TRUE ~ NA\_character\_),

sector = case\_when(sector != "-1" ~ as.character(sector), TRUE ~ NA\_character\_),

lower\_bound\_salary = str\_extract(salary\_estimate,

pattern = "[:digit:]{2,3}"),

lower\_bound\_salary = as.numeric(lower\_bound\_salary) \* 1000,

upper\_bound\_salary = str\_extract(salary\_estimate,

pattern = "([:digit:]{2,})(?=K [\\(G)](file://\\(G)\)"),

upper\_bound\_salary = as.numeric(upper\_bound\_salary) \* 1000,

average\_bound = (lower\_bound\_salary + upper\_bound\_salary) / 2

)

# Salary Distribution of lower bound

ggplot(analyst\_df, aes(x=lower\_bound\_salary)) +

geom\_histogram(fill="white",

col="black")+

labs(y="Count",

title="Distribution of Lower Bound Salary") +

scale\_x\_continuous(name="Lower Bound Salary", limits=c(0, 100000)) +

scale\_y\_continuous(name="Count", limits=c(0, 375)) +

theme\_bw()

# Salary Distribution High End

ggplot(analyst\_df, aes(x=upper\_bound\_salary)) +

geom\_histogram(fill="white",

col="black")+

labs(y="Count",

title="Distribution of Upper Bound Salary") +

scale\_x\_continuous(name="Upper Bound Salary", limits=c(0, 210000)) +

scale\_y\_continuous(name="Count", limits=c(0, 335)) +

theme\_bw()

# Average Data Analyst Salary Distribution

avg\_sal\_dist =

ggplot(analyst\_df, aes(x=average\_bound)) +

geom\_histogram(

fill="white",

col="black")+

labs(x="Average Bound Salary",

y="Count",

title="Average Data Analyst Salary Distribution")+

scale\_x\_continuous(name="Upper Bound Salary", limits=c(0, 150000)) +

scale\_y\_continuous(name="Count", limits=c(0, 350)) +

theme\_bw()

# Salary spread by Sector

sector\_df <- analyst\_df %>%

transmute(

val1 = lower\_bound\_salary,

val2 = average\_bound,

val3 = upper\_bound\_salary,

cat = sector

) %>%

na.omit(sector\_df) %>%

group\_by(cat) %>%

summarise(

min = min(val1, na.rm = T),

avg = mean(val2, na.rm = T),

max = max(val3, na.rm = T)

) %>%

arrange(cat)

ggplot() +

# reshape the data frame & get min value in order to draw first eye-tracking

geom\_segment(

data = gather(sector\_df, measure, val, -cat) %>%

group\_by(cat) %>%

top\_n(-1) %>%

slice(1) %>%

ungroup(),

aes(x = 0, xend = val, y = cat, yend = cat),

linetype = "dotted", size = 0.5, color = "gray80"

) +

# reshape the data frame & get min/max category values to draw segment

geom\_segment(

data = gather(sector\_df, measure, val, -cat) %>%

group\_by(cat) %>%

summarise(start = range(val)[1], end = range(val)[2]) %>%

ungroup(),

aes(x = start, xend = end, y = cat, yend = cat),

color = "gray80", size = 1

) +

# reshape the data frame & plot the points

geom\_point(

data = gather(sector\_df, measure, value, -cat),

aes(value, cat, color = measure),

size = 3

) +

# Adding labels

scale\_x\_comma(position = "bottom", limits = c(0, 200000), labels=scales::dollar\_format()) +

scale\_color\_ipsum(name = "Legend") +

labs(

x = NULL, y = NULL,

title = "Salary Spread by Sector"

) +

theme\_ipsum\_rc(grid = "X") +

theme(legend.position = "top")

# Heat map

ggplot(analyst\_df, aes(long, lat, group = group))+

geom\_polygon(aes(fill = total), color = "white")+

scale\_fill\_distiller(palette= "OrRd", direction=1)+

labs(x="",

y="",

title = "World COVID Deaths",

fill="Number Of Deaths")+

theme\_bw()+

theme(plot.title = element\_text(size=22)

,axis.text.x= element\_text(size=15),

axis.text.y= element\_text(size=15),

axis.title=element\_text(size=18))

**output:**

