**#INTRODUCTION**

**-Visualizing data with factor : (Benefit : yg aka nada si sumbu y hanya yg ada saja tidak semua di plot kan)**

**# Load the ggplot2 package**

**library(ggplot2)**

**# Explore the mtcars data frame with str()**

**str(mtcars)**

**ggplot(mtcars, aes(factor(cyl), mpg)) + geom\_point()**

**-Shape :**

**library(ggplot2)**

**ggplot(mtcars, aes(wt, mpg, shape = factor(cyl))) +**

**geom\_point()**

**\*shape hanya untuk categorical data bukan continous**

**-geom\_smooth() : menambahkan garis persebaran data**

**library(gapminder)**

**library(dplyr)**

**library(ggplot2)**

**ggplot(gapminder, aes(lifeExp, gdpPercap)) +geom\_point() +geom\_smooth()**

**-Alpha : Membuat scatter point menjadi transparan. Nilainya antara 0-1**

**library(gapminder)**

**library(dplyr)**

**library(ggplot2)**

**ggplot(gapminder, aes(lifeExp, gdpPercap, color = continent)) +**

**geom\_point(alpha = 0.5) +geom\_smooth()**

**-Menyimpan plot dalam sebuah function:**

**library(gapminder)**

**library(dplyr)**

**library(ggplot2)**

**life\_Exp\_vs\_gdpPercap <- ggplot(gapminder, aes(lifeExp, gdpPercap))**

**life\_Exp\_vs\_gdpPercap + geom\_point(aes(color = continent)) -> Fungsinya sama, dengan menempatkan aes di awal**

**-Typical Visible Aesthetic :**

**X = X position, Y = Y position, fill = fill color, color, size, alpha = transparency, linetype = line dash pattern, labels = text on plot or axes, shape**

**-Menentukan jenis dan ukuran point :**

**library(datasets)**

**library(dplyr)**

**library(ggplot2)**

**mtcars\_2 <- mtcars %>%**

**mutate(fcyl = as.factor(cyl))**

**ggplot(mtcars\_2, aes(wt, mpg, color = fcyl)) +**

**# Set the shape and size of the points**

**geom\_point(shape = 1, size = 4)**

**-Fill : (Memberikan warna pada inside point, sedangkan color outside/ pinggirannya**

**library(datasets)**

**library(dplyr)**

**library(ggplot2)**

**mtcars\_2 <- mtcars %>%**

**mutate(fcyl = as.factor(cyl),**

**fam = as.factor(am))**

**ggplot(mtcars\_2, aes(wt, mpg, fill= fcyl)) +**

**# Set the shape and size of the points**

**geom\_point(shape = 21, size = 4, alpha = 0.6)**

**-Alpha on Column Name :**

**library(datasets)**

**library(dplyr)**

**library(ggplot2)**

**mtcars\_2 <- mtcars %>%**

**mutate(fcyl = as.factor(cyl),**

**fam = as.factor(am))**

**# Base layer**

**plt\_mpg\_vs\_wt <- ggplot(mtcars\_2, aes(wt, mpg))**

**# Map fcyl to alpha, not size**

**plt\_mpg\_vs\_wt +**

**geom\_point(aes(alpha = fcyl)) -> Jika fcyl kecil maka transparentnya kecil juga dan sebaliknya**

**-Shape on column name :**

**library(datasets)**

**library(dplyr)**

**library(ggplot2)**

**mtcars\_2 <- mtcars %>%**

**mutate(fcyl = as.factor(cyl),**

**fam = as.factor(am))**

**# Base layer**

**plt\_mpg\_vs\_wt <- ggplot(mtcars\_2, aes(wt, mpg))**

**# Map fcyl to alpha, not size**

**plt\_mpg\_vs\_wt +**

**geom\_point(aes(shape = fcyl))**

**-Geom\_text : Mengubah tampilan point di grafik menjadi angka**

**library(datasets)**

**library(dplyr)**

**library(ggplot2)**

**mtcars\_2 <- mtcars %>%**

**mutate(fcyl = as.factor(cyl),**

**fam = as.factor(am))**

**# Base layer**

**plt\_mpg\_vs\_wt <- ggplot(mtcars\_2, aes(wt, mpg))**

**# Map fcyl to alpha, not size**

**plt\_mpg\_vs\_wt +**

**geom\_text(aes(label = fcyl))**

* **Aestethic Label dan Shape bisa untuk categorical data**

**-Coloring with RGB :**

**library(datasets)**

**library(dplyr)**

**library(ggplot2)**

**mtcars\_2 <- mtcars %>%**

**mutate(fcyl = as.factor(cyl),**

**fam = as.factor(am))**

**# A hexadecimal color**

**my\_blue <- "#4ABEFF"**

**# Change the color mapping to a fill mapping**

**ggplot(mtcars\_2, aes(wt, mpg, fill = fcyl)) +**

**# Set point size and shape**

**geom\_point(color=my\_blue, size = 10, shape = 1)**

**-Give the label name to the graph**

**library(datasets)**

**library(dplyr)**

**library(ggplot2)**

**View (mtcars)**

**mtcars\_2 <- mtcars %>%**

**mutate(fcyl = as.factor(cyl),**

**fam = as.factor(am))**

**ggplot(mtcars, aes(wt, mpg, color = fcyl)) +**

**# Add text layer with label rownames(mtcars) and color red**

**geom\_text(label=rownames(mtcars), color="red")**

**-Scale Function :**

* **scale\_x\_\*()**
* **scale\_y\_\*()**
* **scale\_color/colour\_\*()**
* **scale\_fill\_\*()**
* **scale\_shape\_\*()**
* **scale\_linetype\_\*()**
* **scale\_size\_\*()**
* **scale\_x\_continous\_\*()**
* **scale\_color\_discrete\*()**

**-Dodge Position : Membuat grafik batang menjadi side by side**

**library(datasets)**

**library(dplyr)**

**library(ggplot2)**

**mtcars\_2 <- mtcars %>%**

**mutate(fcyl = as.factor(cyl),**

**fam = as.factor(am))**

**palette <- c(automatic = "#377EB8", manual = "#E41A1C")**

**# Set the position**

**ggplot(mtcars\_2, aes(fcyl, fill = fam)) +**

**geom\_bar(position = 'dodge') +**

**labs(x = "Number of Cylinders", y = "Count")**

**scale\_fill\_manual("Transmission", values = palette)**

**-Jitter Position :**

**library(datasets)**

**library(dplyr)**

**library(ggplot2)**

**mtcars\_2 <- mtcars %>%**

**mutate(fcyl = as.factor(cyl),**

**fam = as.factor(am))**

**# Plot 0 vs. mpg**

**ggplot(mtcars, aes(x= mpg, y=0)) +**

**# Add jitter**

**geom\_point(position = 'jitter') +**

**ylim(-2,2) \*Mengatur ukuran sumbu Y**

**#Jitter : untuk men zoom out sumbu y**

**-Tipe Geometries :**

**\*abline, area, bar, bin2d, blank, boxplot, col, contour, count, crossbar, curve, density, density2d, density\_2d, dotplot, errorbar, errorbarh, freqpoly, hex, histogram, hline, jitter, label, line, linerange, map, path, point, pointrange, polygon, qq, qq\_line, quantile, raster, rect, ribbon, rug, segment, sf, sf\_label, sf\_text, smooth, spoke, step, text, tile, violin, vline.**

**-Tipe plot dan possible geoms :**

**\*scatter plots : point, jitter, abline, smooth, count**

**-Using Statistic summary to plot :**

**library(datasets)**

**library(dplyr)**

**library(ggplot2)**

**iris %>%**

**group\_by(Species) %>%**

**summarise\_all(mean) -> iris.summary**

**iris.summary**

**ggplot(iris, aes(Sepal.Length, Sepal.Width, color = Species)) +**

**geom\_point() + -> Untuk melihat persebaran datanya**

**geom\_point(data = iris.summary, shape = 15, size =5) -> Untuk melihat summarynya**

**-Menambahkan stroke pada plot : stroke berfungsi untuk melebarkan lingkaran luar**

**library(datasets)**

**library(dplyr)**

**library(ggplot2)**

**iris %>%**

**group\_by(Species) %>%**

**summarise\_all(mean) -> iris.summary**

**iris.summary**

**ggplot(iris, aes(Sepal.Length, Sepal.Width, color = Species)) +**

**geom\_point() +geom\_point(data = iris.summary, shape = 21, size =5,**

**fill = "black", stroke =2)**

**-Geom Jitter : It adds a small amount of random variation to the location of each point, and is a useful way of handling overplotting caused by discreteness in smaller datasets.**

Overplotting is when one or more points are in the same place (or close enough to the same place) that you can't look at the plot and tell how many points are there.

Geom\_jitter getting to used for small dataset.

**-Geom Jitter :**

**library(ggplot2)**

**library(dplyr)**

**mtcars <- mtcars %>%**

**mutate(fcyl = as.factor(cyl),**

**fam = as.factor(am))**

**# Plot base**

**plt\_mpg\_vs\_fcyl\_by\_fam <- ggplot(mtcars, aes(fcyl, mpg, color = fam))**

**# Default points are shown for comparison**

**plt\_mpg\_vs\_fcyl\_by\_fam + geom\_point() +**

**geom\_point(position= position\_jitter (jitter.width=0.3))**

**-Geom Jitter Dodge :**

**library(ggplot2)**

**library(dplyr)**

**mtcars <- mtcars %>%**

**mutate(fcyl = as.factor(cyl),**

**fam = as.factor(am))**

**# Plot base**

**plt\_mpg\_vs\_fcyl\_by\_fam <- ggplot(mtcars, aes(fcyl, mpg, color = fam))**

**# Default points are shown for comparison**

**plt\_mpg\_vs\_fcyl\_by\_fam + geom\_point() +**

**geom\_point(position = position\_jitterdodge(jitter.width=0.3, dodge.width=0.3))**

**-Geom Jitter jika pakai position = “jitter”**

ggplot(iris, aes(Sepal.Length, Sepal.Width, color = Species)) +

  # Set the position to jitter

  geom\_point(position='jitter',alpha = 0.5)

**-Geom Jitter jika pakai position = position\_jitter**

**# Plot base**

**plt\_mpg\_vs\_fcyl\_by\_fam <- ggplot(mtcars, aes(fcyl, mpg, color = fam))**

**# Default points are shown for comparison**

**plt\_mpg\_vs\_fcyl\_by\_fam + geom\_point() +**

**geom\_point(position = position\_jitterdodge(jitter.width=0.3, dodge.width=0.3))**

**-Histogram : Default bins =30**

**-Histogram with density : Density adalah percentage jumlah data dibagi dengan total data**

**library(ggplot2)**

**library(datasets)**

**library(dplyr)**

**ggplot(mtcars, aes(mpg, ..density..)) +**

**geom\_histogram(binwidth = 1, fill="light blue")**

**-Position in Histogram :**

**\*Stack (Default) : Bars for different groups are stacked on top of each other**

**\*dodge : Bars for different group are placed side by side**

**\*fill : Bars for different groups are shown as proportions**

**\*identity : plot the values as they appear in the datasets**

**-Position in Histogram :**

**library(ggplot2)**

**library(datasets)**

**library(dplyr)**

**ggplot(mtcars, aes(mpg, fill=fam)) +**

**geom\_histogram(binwidth = 1)**

**ggplot(mtcars, aes(mpg, fill = fam)) +**

**# Change the position to dodge**

**geom\_histogram(binwidth = 1, position = 'dodge')**

**ggplot(mtcars, aes(mpg, fill = fam)) +**

**# Change the position to fill**

**geom\_histogram(binwidth = 1, position = "fill")**

**ggplot(mtcars, aes(mpg, fill = fam)) +**

**# Change the position to identity, with transparency 0.4**

**geom\_histogram(binwidth = 1, position = "identity")**

**-Bar Plot :**

**\*stack : the default**

**\*dodge : preffered**

**\*fill : to show proportion**

**-Position in Geom\_Bar :**

**library(ggplot2)**

**library(datasets)**

**library(dplyr)**

**# Plot fcyl, filled by fam**

**ggplot(mtcars, aes(fcyl, fill=fam)) +**

**# Add a bar layer**

**geom\_bar()**

**ggplot(mtcars, aes(fcyl, fill = fam)) +**

**# Set the position to "fill"**

**geom\_bar(position='fill')**

**ggplot(mtcars, aes(fcyl, fill = fam)) +**

**# Change the position to "dodge"**

**geom\_bar(position = "dodge")**

**-**Remember, the reason you want to use position\_dodge() (and position\_jitter()) is to specify how much dodging (or jittering) you want.

**-Position\_dodge :**

**library(ggplot2)**

**library(datasets)**

**library(dplyr)**

**ggplot(mtcars, aes(cyl, fill = fam)) +**

**# Set the transparency to 0.6**

**geom\_bar(alpha=0.6, position = position\_dodge(width = 0.2))**

**-Geom\_Line :**

fish.species contains the global capture rates of seven salmon species from 1950–2010. Each variable (column) is a Salmon species and each observation (row) is one year. fish.tidy contains the same data, but in three columns: Species, Year, and Capture (i.e. one variable per column).

# Plot the Rainbow Salmon time series

ggplot(fish.species, aes(x = Year, y = Rainbow)) +

  geom\_line()

# Plot the Pink Salmon time series

ggplot(fish.species, aes(x = Year, y = Pink)) +

  geom\_line()

# Plot multiple time-series by grouping by species

ggplot(fish.tidy, aes(Year, Capture)) +

  geom\_line(aes(group = Species))

# Plot multiple time-series by coloring by species

ggplot(fish.tidy, aes(x = Year, y = Capture, color = Species)) +

  geom\_line(aes(group = Species))

**-Linetype Aestethic in Geom Line :**

**ggplot(fish, aes(Year, Capture, linetype = Species)) + geom\_line()**

**-Geom\_area : membuat line area dengan garis warna di bawah**

**ggplot(fish, aes(Year, Capture, fill = Species)) + geom\_area()**

**-Geom\_area with fill position :**

**ggplot(fish, aes(Year, Capture, fill = Species)) + geom\_area(position = “fill”)**

**-Geom\_ribbon :**

**ggplot(fish, aes(Year, Capture, fill = Species)) + geom\_ribbon(aes(ymax = Capture, ymin =0), alpha = 0.3)**

**-Menagtur ukuran font dalam grafik :**

**Plot(Age, Height, main = “Scatterplot”, cex=0.5) -> mengatur ukuran point di grafiknya**

**Plot(Age, Height, main =”Scatterplot”, cex.main=20) -> Ukuran Judul**

**Plot(Age, Height, main =”Scatterplot”, cex.lab = 15) -> Ukuran Label**

**Plot(Age, Height, main =”Scatterplot”, cex.axis = 0.7) -> ukuran axis**

**-Subset Plot :**

**Plot (Age[Gender==”Male”], Height[Gender==”Male”])**

**#Theme**

**-The Text element :**

**Theme(**

**Text**

**axis.title**

**axis.title.x**

**Axis.title.x.top**

**Axis.title.x.bottom**

**Axis.title.y**

**Axis.title.y.left**

**Axis.title.y.right**

**Title**

**Legend.title**

**Plot.title**

**Plot.subtitle**

**Plot.caption**

**Plot.tag**

**Axis.text**

**Axis.text.x**

**Axis.text.x.top**

**Axis.text.x.bottom**

**Axis.text.y**

**Axis.text.y.left**

**Axis.text.y.right**

**Legend.text**

**Strip.text**

**Strip.text.x**

**Strip.text.y**

**-Mewarnai label :**

**Ggplot + theme(axis.title = element\_text(color = “blue”)**

**-Line Element :**

**Theme(**

**Line,**

**Axis.ticks,**

**Axis.ticks.x,**

**Axis.ticks.x.top,**

**Axis.ticks.x.bottom,**

**Axis.ticks.y,**

**Axis.ticks.y.left,**

**Axis.ticks.y.right,**

**Axis.line,**

**Axis.line.x,**

**Axis.line.x.top,**

**Axis.line.x.bottom,**

**Axis.line.y,**

**Axis.line.y.left,**

**Axis.line.y.right,**

**Panel.grid,**

**Panel.grid.major,**

**Panel.grid.major.x,**

**Panel.grid.major.y,**

**Panel.grid.minor,**

**Panel.grid.minor.x,**

**Panle.grid.minor.y)**

**-Rect elements :**

**Theme(**

**Rect,**

**Legend.background,**

**Legend.key,**

**Legend.box.background,**

**Panel.background,**

**Panel.border,**

**Plot.background,**

**Strip.background,**

**Strip.background.x,**

**Strip.background.y)**

**-element\_blank() : membuat semua line, rect dan text hilang**

**Ggplot + theme(line = element\_blank(),**

**Rect = element\_blank(),**

**Text = element\_blank())**

**-Moving The legend :**

**P+ theme(legend.position = new\_value)**

* **”top”, “bottom”, “left”, “right” : place it at the side of the plot**
* **“none”**
* **C(x,y) : c(0,0) means the bottom left, c(1,1) means the top right**

**P+ theme(legend.position = c(0.5,0.5))**

**-Mewarnai garis kolom :**

**P+theme(axis.line = element\_line(color ="red", linetype = "dashed"))**

**\*axis.line.x : mewarnai garis x**

**\*axis.line.y : mewarnai garis y**

**\*axis.line : mewarnai seluruh garis bagian luar**

**-Mewarnai seluruh grafik :**

**P+ theme(**

**# For all rectangles, set the fill color to grey92**

**rect = element\_rect(fill = "grey92"),**

**# For the legend key, turn off the outline**

**legend.key = element\_rect(color = NA))**

**-Remove Axis ticks and panel grid :**

**P+ theme(**

**# For all rectangles, set the fill color to grey92**

**rect = element\_rect(fill = "grey92"),**

**# For the legend key, turn off the outline**

**legend.key = element\_rect(color = NA)**

**# Turn off axis ticks**

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

**# Turn off the panel grid**

**panel.grid = element\_blank())**

**-Menambahkan warna garis Y pada grid :**

**P+ theme(**

**# For all rectangles, set the fill color to grey92**

**rect = element\_rect(fill = "grey92"),**

**# For the legend key, turn off the outline**

**legend.key = element\_rect(color = NA)**

**# Turn off axis ticks**

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

**# Turn off the panel grid**

**panel.grid = element\_blank(),**

**# Add major y-axis panel grid lines back**

**panel.grid.major.y = element\_line(**

**# Set the color to white**

**color = "white",**

**# Set the size to 0.5**

**size = 0.5,**

**# Set the line type to dotted**

**linetype = "dotted"**

**)**

**)**

**-Mengatur ukuran teks dan judul :**

**theme(**

**# For all rectangles, set the fill color to grey92**

**rect = element\_rect(fill = "grey92"),**

**# For the legend key, turn off the outline**

**legend.key = element\_rect(color = NA),**

**# Turn off axis ticks**

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

**# Turn off the panel grid**

**panel.grid = element\_blank(),**

**# Add major y-axis panel grid lines back**

**panel.grid.major.y = element\_line(**

**# Set the color to white**

**color = "white",**

**# Set the size to 0.5**

**size = 0.5,**

**# Set the line type to dotted**

**linetype = "dotted"**

**),axis.text = element\_text(color = "grey25"),**

**# Set the plot title font face to italic and font size to 16**

**plot.title=element\_text(size = 16, face ="italic")**

**)**

**-Unit pada white space :**

**unit(x, unit)**

**x=number**

**unit = satuan**

**-setting border :**

**Margin(top, right, bottom, left, unit)**

**\*default unit is pt**

**-Setting unit axis.ticks.length :**

**P + theme(axis.ticks.length = unit(2, "lines")**

**)**

**-Setting Legend Unit :**

**P + theme(**

**# Set the legend key size to 3 centimeters**

**legend.key.size = unit (3,"cm"))**

**-Setting Margin :**

**P+ theme(**

**# Set the legend margin to (20, 30, 40, 50) points**

**legend.margin = margin(20, 30, 40, 50, "pt")**

**)**

**-Defining theme objects**

**Z <- ggplot(iris, aes(Sepal.Length, Sepal.Width, color = Species)) +**

**Scale\_x\_continuous(“Sepal Length (cm)”, limits = c(4,8), expand = c(0,0)) +**

**Scale\_y\_continuous(“Sepal Width (cm)”, limits = c(1.5, 5.5), expand = c(0,0)) +**

**Scale\_color\_brewer(“Species”, palette = “Dark2”, labels = c(“Setosa”, “Versicolor”, “Virginica”))**

**Z+ theme(text = element\_text(family = “serif”, size = 14),**

**Rect = element\_blank(),**

**Panel.grid = element\_blank(),**

**Title = element\_text(color = “#8b0000”),**

**Axis.line = element\_line(color = “black”))**

**-Setting Themes :**

**Original <- theme(text = element\_text(family = “serif”, size = 14),**

**Rect = element\_blank(),**

**Panel.grid = element\_blank(),**

**Title = element\_text(color = “#8b0000”),**

**Axis.line = element\_line(color = “black”))**

**Theme\_set(original)**

**-Template Themes :**

* [**theme\_gray()**](http://www.rdocumentation.org/packages/ggplot2/functions/ggtheme) is the default.
* [**theme\_bw()**](http://www.rdocumentation.org/packages/ggplot2/functions/ggtheme) is useful when you use transparency.
* [**theme\_classic()**](http://www.rdocumentation.org/packages/ggplot2/functions/ggtheme) is more traditional.
* [**theme\_void()**](http://www.rdocumentation.org/packages/ggplot2/functions/ggtheme) removes everything but the data.

**-Add Theme\_bw() :**

**ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width, color = Species)) +**

**geom\_point() + theme\_bw()**

**-Combining Theme :**

# Theme layer saved as an object, theme\_recession

theme\_recession <- theme(

  rect = element\_rect(fill = "grey92"),

  legend.key = element\_rect(color = NA),

  axis.ticks = element\_blank(),

  panel.grid = element\_blank(),

  panel.grid.major.y = element\_line(color = "white", size = 0.5, linetype = "dotted"),

  axis.text = element\_text(color = "grey25"),

  plot.title = element\_text(face = "italic", size = 16),

  legend.position = c(0.6, 0.1)

)

# Combine the Tufte theme with theme\_recession

theme\_tufte\_recession <- theme\_tufte() + theme\_recession

# Add the recession theme to the plot

plt\_prop\_unemployed\_over\_time + theme\_tufte\_recession

**-Setting the theme :**

theme\_recession <- theme(

  rect = element\_rect(fill = "grey92"),

  legend.key = element\_rect(color = NA),

  axis.ticks = element\_blank(),

  panel.grid = element\_blank(),

  panel.grid.major.y = element\_line(color = "white", size = 0.5, linetype = "dotted"),

  axis.text = element\_text(color = "grey25"),

  plot.title = element\_text(face = "italic", size = 16),

  legend.position = c(0.6, 0.1)

)

theme\_tufte\_recession <- theme\_tufte() + theme\_recession

# Set theme\_tufte\_recession as the default theme

theme\_set <- theme\_tufte\_recession

# Draw the plot (without explicitly adding a theme)

plt\_prop\_unemployed\_over\_time

**-Efficienct Plot :**

**library(ggplot2)**

**library(gapminder)**

**library(dplyr)**

**gm2007 <- gapminder %>%**

**filter(year==2007) %>%**

**arrange(desc(lifeExp))**

**gm2007**

**gm2007\_head <- head(gm2007,10)**

**gm2007\_head**

**gm2007\_tail <- tail(gm2007,10)**

**gm2007\_tail**

**gm2007\_all <- rbind(gm2007\_head, gm2007\_tail)**

**gm2007\_all**

# Set the color scale

palette <- brewer.pal(5, "RdYlBu")[-(2:4)]

# Add a title and caption

ggplot(gm2007, aes(x = lifeExp, y = country, color = lifeExp)) +

  geom\_point(size = 4) +

  geom\_segment(aes(xend = 30, yend = country), size = 2) +

  geom\_text(aes(label = round(lifeExp,1)), color = "white", size = 1.5) +

  scale\_x\_continuous("", expand = c(0,0), limits = c(30,90), position = "top") +

  scale\_color\_gradientn(colors = palette) +

  labs(title = "Highest and lowest life expectancies, 2007", caption = "Source: gapminder")

**-Use Annotate :**

# Add a curve

plt\_country\_vs\_lifeExp +

  step\_1\_themes +

  geom\_vline(xintercept = global\_mean, color = "grey40", linetype = 3) +

  step\_3\_annotation +

  annotate(

    "curve",

    x = x\_start, y = y\_start,

    xend = x\_end, yend = y\_end,

    arrow = arrow(length = unit(0.2, "cm"), type = "closed"),

    color = "grey40"

  )