08/04/25 Week-12

```
#Install necessary packages
install.packages(c("dplyr", "ggplot2", "gapminder"))
library(dplyr)
library(ggplot2)
library(gapminder)
#Load the Gapminder dataset
data(gapminder)
head(gapminder)
```

#A tibble : 6×6

Country <fct>	Continent <fct>	year <int>	lifeExp <dbl>	pop <int>	gdpPercap <dbl>
1. Afghanistan	Asia	1952	28.8	8425333	779
2. Afghanistan	Asia	1957	30.3	9240934	821
3. Afghanistan	Asia	1962	32.0	10267083	853
4. Afghanistan	Asia	1967	34.0	11537966	836
5. Afghanistan	Asia	1972	36.1	13079460	740
6. Afghanistan	Asia	1977	38.4	14880372	786

```
summary(gapminder)
```

Country	Continent	year	lifeExp	pop
Afghanistan: 12	Africa: 624	Min. :1952	Min. :23.60	Min. 6001
Albania: 12	Americas: 300	1st Qu.:1966	1st Qu.:48.20	1st Qu.: 2.794e
Algeria: 12	Asia: 396	Median :1980	Median :60.71	Median : 7.024e06
Angola: 12	Europe: 360	Mean :1980	Mean :59.47	Mean : 2.960e07
Argentina: 12	Oceania: 24	3rd Qu.:1993	3rd Qu.:70.85	3rd Qu.: 1.958e07
Australia: 12		Max. :2007	Max. :82.60	Max. 1.319e09

(olles): 1632
gdpPercap
Min : 241.2
1st qu : 1202.1
median : 3531.8
mean : 7215.8
3rd qu: 9325.5
Max : 113523.1

gapminder-2007 <- filter (gapminder, year == 2007)
head (gapminder-2007)

A tibble : 6 × 6

Country <fct>	continent <fct>	year <int>	lifeExp <dbl>	pop <int>	gdpPercap <dbl>
Afghanistan	Asia	2007	43.8	31889923	925.
Albania	Europe	2007	76.4	3600523	5937.
Algeria	Africa	2007	72.3	33333216	6223
Angola	Africa	2007	42.7	12420476	4797
Argentina	Americas	2007	75.3	40301927	12779
Australia	Oceania	2007	81.2	20434176	34435

life_expectancy_by_continent <- gapminder %>%
group_by (continent) %>%
summarize (avg_life_exp = mean (lifeExp, na.rm = TRUE))

head (life_expectancy_by_continent)

A tibble : 5 × 2

continent <fct>	avg_life_exp <dbl>
Africa	48.9
Americas	64.7
Asia	60.1
Europe	71.9
Oceania	72.3

cor(gapminder$gdpPercap, gapminder$lifeExp,
use = "complete.obs")

[1] 0.5833062

model ← lm(lifeExp ~ gdpPercap, data = gapminder)

summary(model)

Call:
lm(formula = lifeExp ~ gdpPercap, data = gapminder)

Residuals

Min	1Q	Median	3Q	Max
-82.754	-7.758	2.176	8.225	18.426

Coefficients:

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	5.396e+01	3.150e-01	171.29	<2e-16 ***
gdpPercap	7.649e-04	2.579e-05	29.66	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*'
0.05 '.' 0.1 ' '

Residual standard error: 10.49 on 1702 degrees of freedom

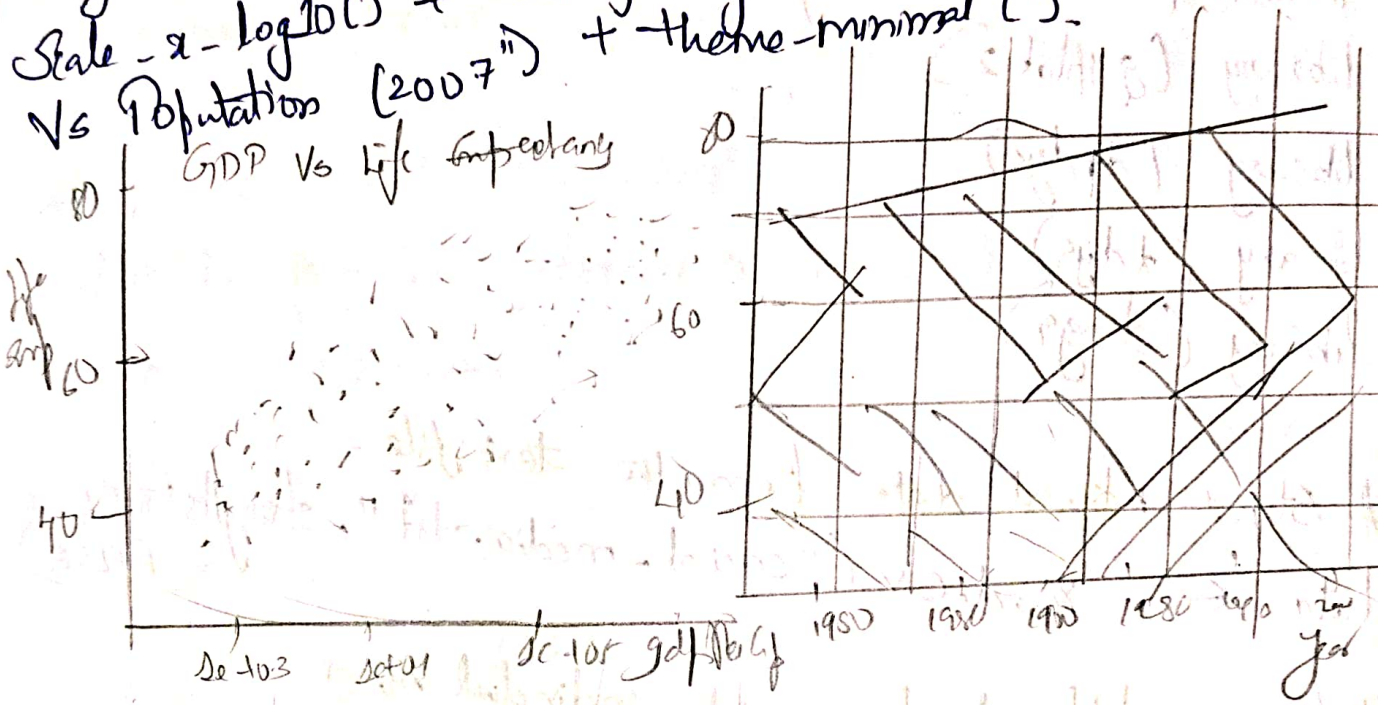
Multiple R-squared: 0.3407, Adjusted R-squared: 0.3403

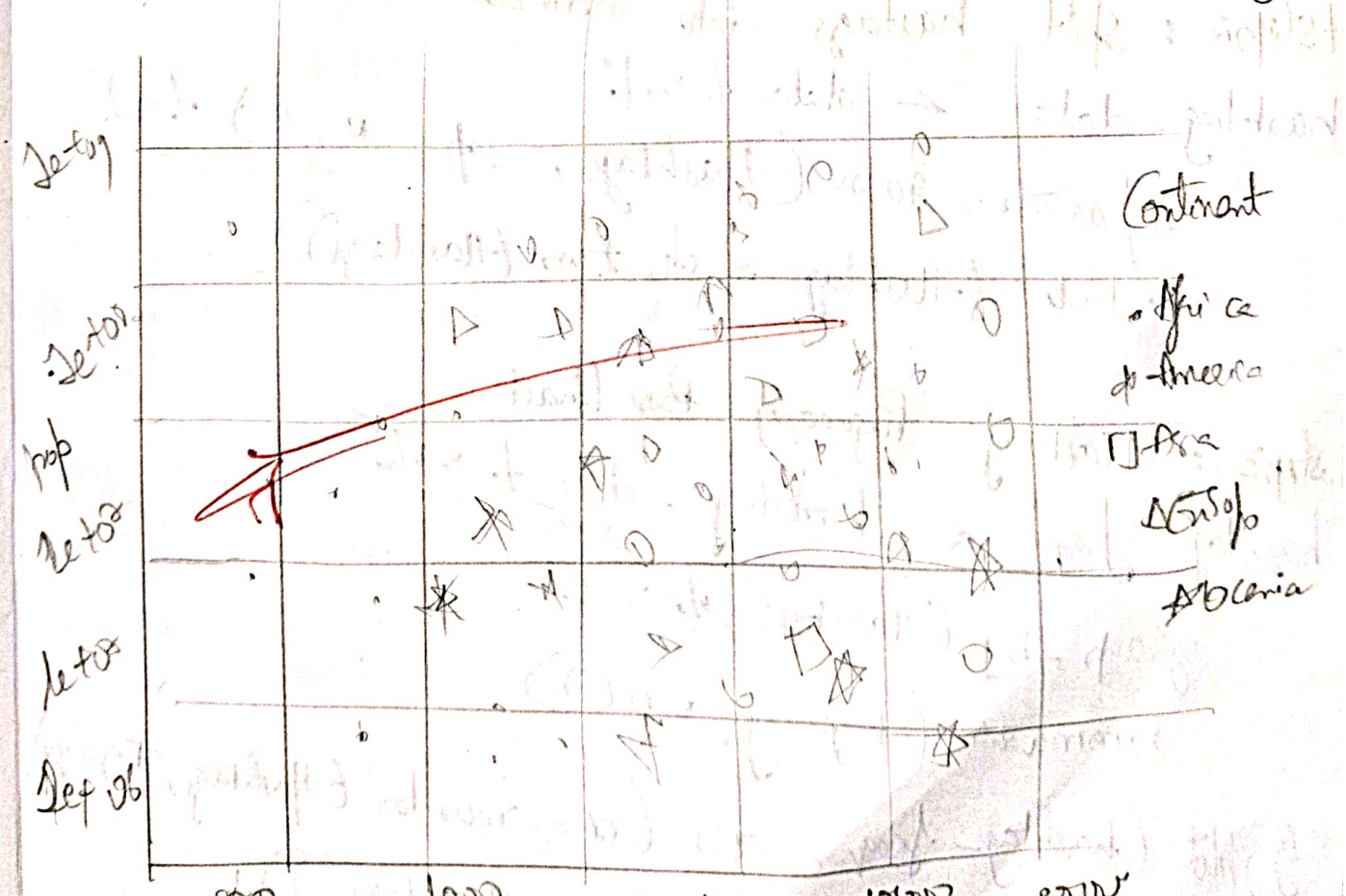
F-statistic: 879.6 on 1 and 1702 DF, p-value: <2.2e-16

ggplot(gapminder, aes(x = gdpPercap, y = lifeExp)) +
geom_point(aes(color = Continent), alpha = 0.7)) +
scale_x_log10() + labs(title = "GDP Vs Life Expectancy") +
theme_minimal()

ggplot(gapminder, aes(x=year, y=lifeExp, color=continent)) +
geom_line() + labs(title="Life Expectancy over time") +
theme_minimal().

gapminder_2007 <- filter(gapminder, year==2007)
ggplot(gapminder_2007, aes(x=gdpPercap, y=pop)) +
geom_point(aes(color=continent), alpha=0.7) +
scale_x_log10() + scale_y_log10() + labs(title="GDP
Vs Population (2007)") + theme_minimal().





15/04/25

Week - 13

Use R-Project for data visualization of social media data -

```
# Load required libraries
install.packages("tidyr")
library(ggplot2)
library(dplyr)
library(tidyr)
library(stringr)

# Step 1 : Read data from the text file -
data <- read.csv("social_media.txt", stringsAsFactors = FALSE)

#Step 2 : Split hashtags into individual rows
hashtag_data <- data %>%
  separate_rows(Hashtags, sep = ",") %>%
  mutate(Hashtags = str_trim(Hashtags))

#Step 3 : Hashtag Frequency Bar Chart
hashtag_freq <- hashtag_data %>%
  group_by(Hashtags) %>%
  summarise(Frequency = n())

ggplot(hashtag_freq, aes(x = reorder(Hashtags, Frequency),
    y = Frequency, fill = Hashtags)) +
```

```
geom_bar(stat = "identity") +
coord_flip() +
labs(title = "Hashtag frequency", x = "Hashtag",
     y = "Count") +
theme_minimal()
```

Step 4 : Sentiment Pie Chart

```
sentiment_freq <- data %>%
  count(Sentiment)

ggplot(sentiment_freq, aes(x = "", y = n, fill = Sentiment)) +
  geom_col(width = 1) +
  coord_polar(theta = "y") +
  labs(title = "Tweet Sentiment Distribution") +
  theme_void().
```

Step 5 : Scatter plot of Likes vs. Retweets

```
ggplot(data, aes(x = Likes, y = Retweets,
                 color = Sentiment)) +
  geom_point(size = 4) +
  labs(title = "Engagement : Likes vs. Retweets",
       x = "Likes", y = "Retweets") +
  theme_minimal().
```

: Social-media.txt

Username, Tweet Text, Hashtags, Likes, Retweets, Sentiment

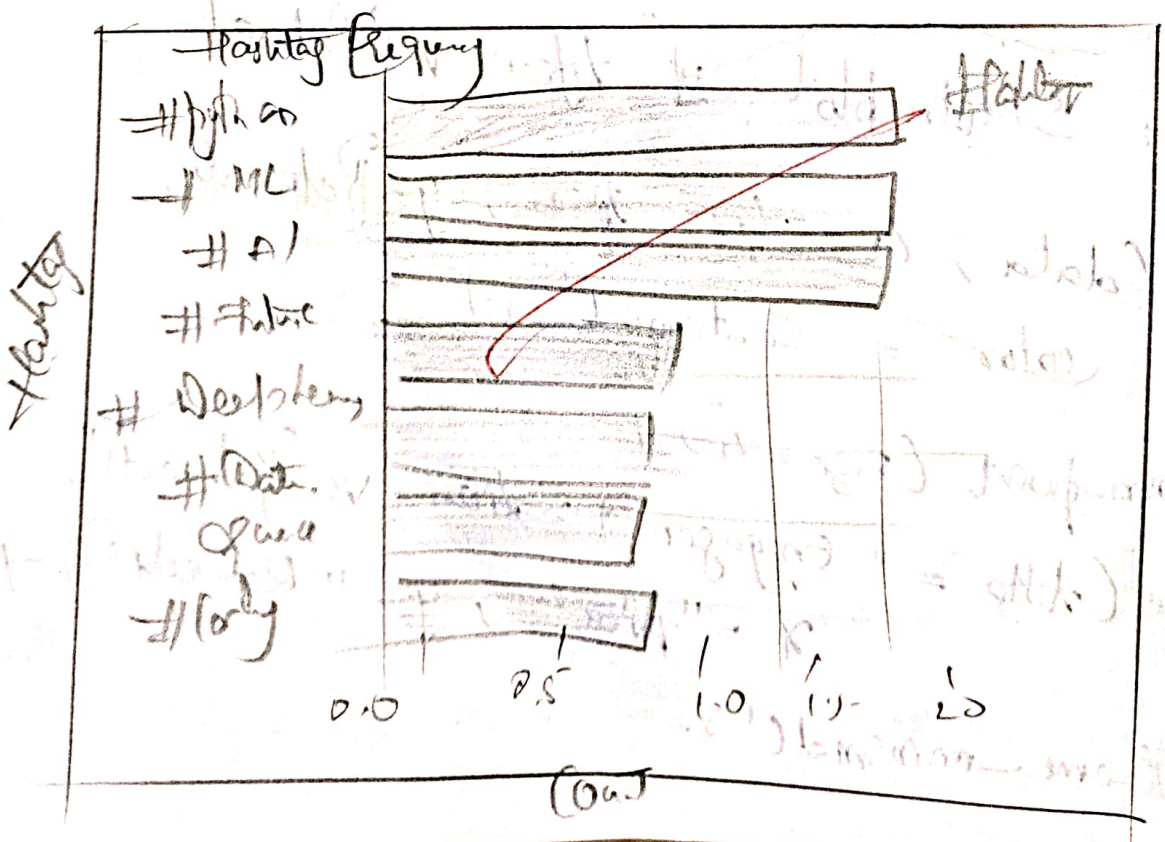
user 1, AI is transforming the future! #AI, #Future, 120, 30, Positive

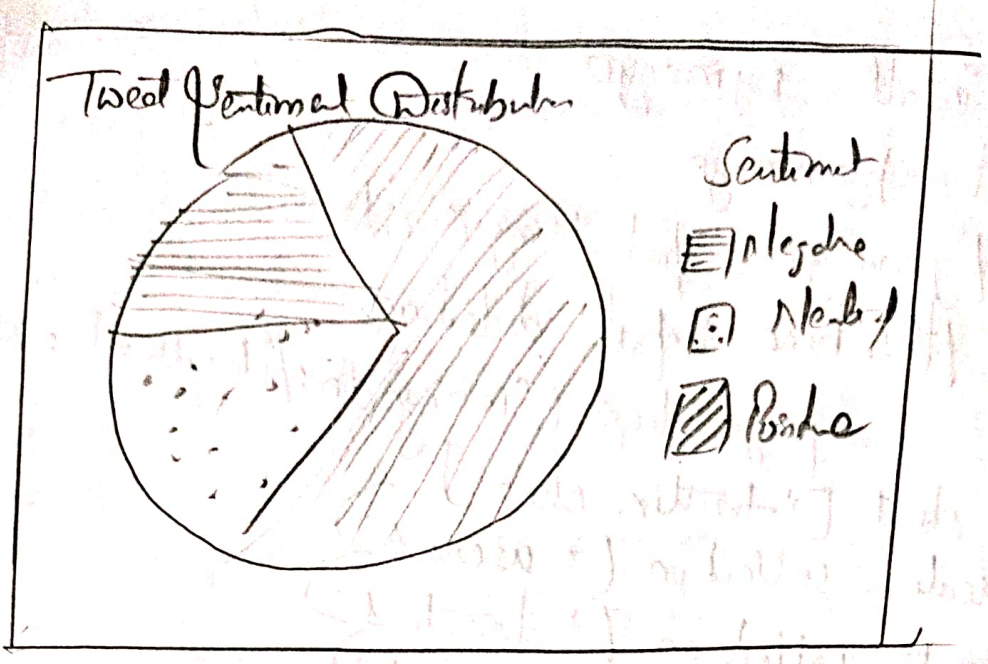
user 2, Struggling with data preprocessing, #Python, 75, 15, Neutral.

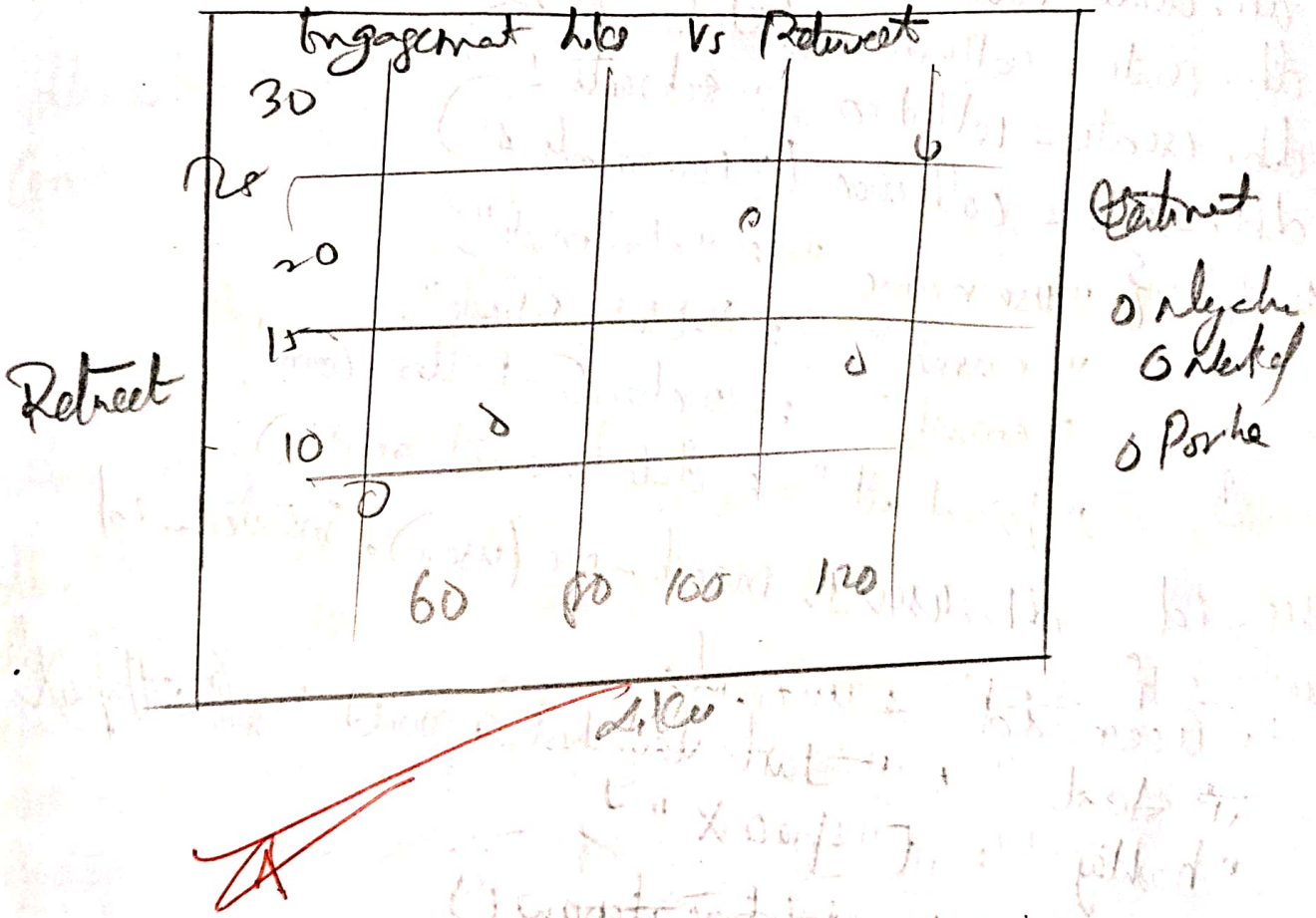
user 3, Machine learning is so easy! #ML, #AI, 95, 25, Positive.

= user 4, Overfitting is real issue, #ML, #Deeplearning, 60, 20, Negative.

= user 5, Python makes coding easier, #Python, #Coding, 110, 20, Positive.




Tweet Sentiment Distribution
Sentiment
Negative
Neutral
Positive


Engagement Likes Vs Retweet
30
25
20
15
10
Retweet
60
80
100
120
Likes
Sentiment
Negative
Neutral
Positive