• code for plots on lat memo.

STATISTICAL ANALYSIS.

R - notes.

Statistical Analysis (TUT 3) (TUT 4).

- tranform numerical to words
  %>% mutate() function

1. basic stats analysis.

① Basic Summary summary()
② Find missing values colsum(is.na())
③ Remove rows na.omit()
④ Summary after clean
⑤ Ensure no n.a. values" colsum(is.na())

⑥ Impute missing values with column means

airquality_impute = airquality %>% mutate(across(everything()
, ~ ifelse(is.na(.), mean(., na.rm = TRUE), .)))

⑦ Replace actual data with imputed data.

⑧ Create Histogram. (each variable except month and day)

⑨ Boxplot by month => Ozone
Boxplot by month => Temp
Boxplot by month => Solar.R
Boxplot by month each variable.

⑩ Scatterplot with smoothing

⑪ Pairwise Scatterplot matrix

Linear regression model:

```
model = lm (Ozone ~ Temp + Wind + Solar.R, data = airq.)
Summary (model)
```

⑫ Timeseries plot (Ozone over time (index))

```
airquality $ DayIndex = 1 : nrow(airquality)
ggplot (airquality, aes(x = daytime, y = Ozone)) +
geom_line (color = "darkgreen") +
geom_smooth (method = "loess") +
labs (title = "  ", x = "  ", y = "  ") +
theme_minimal()
```

8. Facet Scatter plot.

```
ggplot (airquality , aes (x = Temp, y = Ozone)) +
geom_point (aes(color = month)) +
geom_smooth (method = "lm" , se = FALSE) +
Facet_wrap (~ month) +
theme_minimal() +
labs (title = "  ")
```

Statistical analysis:

- Load data

```
date = read.csv(" ", header = T)
```

- Scatterplot of GDP vs unemployed
- ggpairsplot.

- Fit a linear regression model.

```
model = lm(unemployment_rate ~ GDP_per_capita, data = da
summary(model)
```

- Correlation Analysis.

```
x <- ggpairs(mtcars[, c(" ", " ", "b", " ")],
        title = " ")
```

- Print(p5).

Linear regression:

```
model <- lm(mpg ~ hp, data = mtcars)
summary(model)
```

* Extract residuals and fitted values

```
residuals = resid(model)
fitted <- fitted(model)
```

(tut4)

- residuals vs fitted values plot
- normal Q-Q plot
- Scale-Location Plot (spread vs fitted).
- Histogram and density of residuals

Shapiro-Walk Normality test (includes log transform)
code (Tut4 + memo)

TUT3 (code for following plots):

- * Histogram
- * Boxplots
- * Scatterplot with smoothing
- * Pairwise scatterplot matrix
- * Time Series plot.
- * Faceted Scaterplot

} "airquality" dataset used in TUT 3.

- * Scatterplot of GDP vs unempl
- * ggpairs.

} economics dataset from BB.

TUT4 (code for following plots:

- * Histogram (horsepower)
- * Boxplot of MPG grouped by cylinder
- * Density plot of MPG
- * Pairwise plot.
- * Residensual vs fitted model plot
- * Scale location plot.
- * Histogram & Density of residuals.
- * normal Q-Q Plot.

} mtcars dataset.

SVM - models (cut 5)

```
rm (list = ls())
library (AER)
library (e1071)
```

① Load and preview data set:

```
data ("Affairs")
head(    )
```

② Create BINARY RESPONSE variable.

```
Affairs$had_affairs <- as.factor (ifelse (Affairs$affairs > 0, 1, 0))
data = Affairs[, -1]
str (data)
```

③ Manually train-split (70:30)

```
set.seed (100)
n <- nrow(data)

train_index <- sample (1:n, size = 0.7 * n)
train_data <- data[train_index, ]
test_data <- data [-train_index, ]
```

④ Train svm model with linear kernel.

```
svm_model = svm (had_affair ~ ., data = train_data,
                 kernel = "linear", scale = TRUE,
                 probability = "TRUE")
```

⑤ Predict on test data

```
pred_test <- predict(svm_model, newdata = test_data)
```

⑥ Calculate training error

```
training_error <- mean(pred_test != test_data$hod_affair)
print(training_error).
```

⑦ Evaluate and comment on the prediction accuracy of both models.

~~prediction1 <- p~~

```
svm_accuracy1 <- postResample(pred1_test,
                              mtcars_test$mpg).
```

Scaling the data:

```
mtcars_scaled = as.data.frame(scale(mtcars)).
```

Plot tutering multiple line charts.
(Assignment 4a)

```
rm(list = ls())
library(tidyr)
library(ggplot2)

data = read.csv("                ", header = TRUE)
```

① Melt the dataset to a long format for plotting

```
data-long = gather(data, key = "time", value = "value",
  -dummy - variable, -id)

data-long $ time <- as.numeric(as.character(gsub("v",
    data- long $ time)))
```

② Plot using ggplot with facet-wrap.

```
p = ggplot(data-long, aes(x = __ , y = __ , group = id, color
    = dummy- variable)) +
  geom- line(alpha = 0.7) +        set alpha for better
  labs(title = "    ", x = "    ", y = "    ")        visualization.
  Scale- color- manual(values = c("T1" = "darkblue", "T2"
    , "T3" = "red")) +        colour by dummy variable
  theme- classic() +
  theme(legend.title = element-blank()) modify leng
print(p)
```

ANOVA MODEL:

```
model <- aov(weight ~ group, data = PlantGrowth)
summary(model)
```

NORMALISING The Data.

```
maxs <- apply(rock, 2, max)
mins <- apply(min, 2, min)
Scaled-rock <- as.data.frame(scale(rock,
    (centre = mins, scale = maxs - mins))
```

2 amount of variables.