Cheat Sheet for comprehensive Coursera Data Science Specialization by Johns Hopkins University

1. Introduction to Data Science

- **Data Science Definition**: Interdisciplinary field using scientific methods, processes, algorithms, and systems to extract knowledge and insights from structured and unstructured data.

- **Data Science Lifecycle**:

  - **1. Discovery**: Initial research and business understanding.

  - **2. Data Preparation**: Data collection, data cleaning, and data annotation.

  - **3. Model Planning**: Determining the data model and methods.

  - **4. Model Building**: Data analysis and model development.

  - **5. Communication Results**: Visualization and reporting.

  - **6. Operationalize**: Finalizing code, updating models, and monitoring performance.

2. R Programming

- **Basic Syntax**:

  - **Comments**: `# This is a comment`

  - **Variables**: `x <- 5`

  - **Data Types**: `numeric`, `character`, `logical`, `factor`, `integer`, `complex`

  - **Vectors**: `c(1, 2, 3)`

  - **Matrices**: `matrix(data, nrow, ncol, byrow=FALSE)`

  - **Data Frames**: `data.frame(col1, col2)`

  - **Lists**: `list(a=1, b="hello", c=c(1,2,3))`

- **Control Structures**:

  - **If-Else**:

```
if (condition) {
  # code
} else {
```

```
  # code
}
```

- **For Loops**:

```
for (i in 1:10) {
  # code
}
```

- **While Loops**:

```
while (condition) {
  # code
}
```

- **Functions**:

- **Defining Functions**:

```
my_function <- function(arg1, arg2) {
  # code
  return(result)
}
```

- **Anonymous Functions**: `function(x) x^2`

- **Data Manipulation**:

- **dplyr Package**:

- `filter()`: `filter(df, condition)`
- `select()`: `select(df, col1, col2)`
- `mutate()`: `mutate(df, new_col = col1 + col2)`
- `arrange()`: `arrange(df, col1)`
- `summarize()`: `summarize(df, mean_col = mean(col1))`
- `group_by()`: `group_by(df, col1)`

- **Reading and Writing Data**:

- **CSV Files**:

- `read.csv()`: `df <- read.csv("file.csv")`

- `write.csv()`: `write.csv(df, "file.csv")`

- **Excel Files**:

- `read_excel()`: `df <- read_excel("file.xlsx")`
- `write_xlsx()`: `write_xlsx(df, "file.xlsx")`

3. Getting and Cleaning Data

- **Data Sources**:

- **Web Scraping**: `rvest` package

- `read_html()`: `page <- read_html("http://example.com")`
- `html_nodes()`: `nodes <- html_nodes(page, "div")`
- `html_text()`: `text <- html_text(nodes)`

- **APIs**: `httr` package

- `GET()`: `response <- GET("http://api.example.com/data")`
- `content()`: `data <- content(response, "parsed")`

- **Data Cleaning**:

- **Handling Missing Values**:

- `is.na()`: `is.na(df$col)`
- `na.omit()`: `df <- na.omit(df)`
- `complete.cases()`: `df[complete.cases(df), ]`

- **Data Transformation**:

- `mutate()`: `df <- mutate(df, new_col = col1 * 2)`
- `rename()`: `df <- rename(df, new_name = old_name)`
- `gather()`: `df <- gather(df, key, value, -col1)`
- `spread()`: `df <- spread(df, key, value)`

- **Data Merging**:

- **Joins**:

- `inner_join()`: `inner_join(df1, df2, by="key")`
- `left_join()`: `left_join(df1, df2, by="key")`
- `right_join()`: `right_join(df1, df2, by="key")`
- `full_join()`: `full_join(df1, df2, by="key")`

4. Exploratory Data Analysis (EDA)

- **Descriptive Statistics**:

- **Summary**: `summary(df)`
- **Mean**: `mean(df$col, na.rm=TRUE)`
- **Median**: `median(df$col, na.rm=TRUE)`
- **Mode**: `names(sort(-table(df$col)))[1]`
- **Standard Deviation**: `sd(df$col, na.rm=TRUE)`
- **Variance**: `var(df$col, na.rm=TRUE)`

- **Visualization**:
  - **Base R**:

- `plot()`: `plot(df$col1, df$col2)`
- `hist()`: `hist(df$col)`
- `boxplot()`: `boxplot(df$col)`

  - **ggplot2 Package**:

- `ggplot()`: `ggplot(df, aes(x=col1, y=col2)) + geom_point()`
- `geom_histogram()`: `ggplot(df, aes(x=col)) + geom_histogram()`
- `geom_boxplot()`: `ggplot(df, aes(x=col)) + geom_boxplot()`

- **Correlation Analysis**:
  - **Correlation Matrix**: `cor(df)`
  - **Correlation Plot**: `corrplot(cor(df))`

title: "Report Title"

author: "Author Name"

date: "2023-10-01"

output: html_document

```
- **Inline Code**: `` `r code` ``
- **Chunks**:


  code
```

```
  - **Output Formats**: `html_document`, `pdf_document`,
`word_document`

- **Knitr**:
  - **Chunk Options**:
    - `echo=FALSE`: ```` ```{r echo=FALSE} ````
    - `results="hide"`: ```` ```{r results="hide"} ````
    - `fig.width=6`: ```` ```{r fig.width=6} ````

### **6. Statistical Inference**

- **Probability Distributions**:
  - **Normal Distribution**: `dnorm()`, `pnorm()`, `qnorm()`, `rnorm()`
  - **Binomial Distribution**: `dbinom()`, `pbinom()`, `qbinom()`,
`rbinom()`
  - **Poisson Distribution**: `dpois()`, `ppois()`, `qpois()`,
`rpois()`

- **Hypothesis Testing**:
  - **t-Test**: `t.test(df$col1, df$col2)`
  - **Chi-Square Test**: `chisq.test(df$col1, df$col2)`
  - **ANOVA**: `aov(col1 ~ col2, data=df)`

- **Confidence Intervals**:
  - **Mean CI**: `t.test(df$col)$conf.int`
  - **Proportion CI**: `prop.test(x, n)$conf.int`

### **7. Regression Models**

- **Simple Linear Regression**:
  - **Model**: `lm(y ~ x, data=df)`
  - **Summary**: `summary(model)`
  - **Predictions**: `predict(model, newdata)`

- **Multiple Linear Regression**:
  - **Model**: `lm(y ~ x1 + x2, data=df)`
  - **Interaction Terms**: `lm(y ~ x1 * x2, data=df)`

- **Logistic Regression**:
  - **Model**: `glm(y ~ x, data=df, family=binomial)`
  - **Predictions**: `predict(model, newdata, type="response")`

- **Model Diagnostics**:
  - **Residuals**: `residuals(model)`
  - **Residual Plots**: `plot(model)`
  - **Influential Points**: `influence.measures(model)`

### **8. Practical Machine Learning**
```

```
- **Model Training**:
  - **Train-Test Split**:
```

```
set.seed(123)

train_index <- sample(1:nrow(df), 0.7*nrow(df))

train <- df[train_index, ]

test <- df[-train_index, ]
```

```
  - **Cross-Validation**: `trainControl(method="cv", number=10)`

- **Model Types**:
  - **Decision Trees**: `rpart(y ~ x, data=train)`
  - **Random Forest**: `randomForest(y ~ x, data=train)`
  - **Support Vector Machines**: `svm(y ~ x, data=train)`
  - **K-Nearest Neighbors**: `knn(train, test, cl, k)`

- **Model Evaluation**:
  - **Confusion Matrix**: `confusionMatrix(predictions, test$y)`
  - **Accuracy**: `mean(predictions == test$y)`
  - **ROC Curve**: `roc(test$y, predictions)`

### **9. Developing Data Products**

- **Shiny Apps**:
  - **Basic Structure**:
```

```
library(shiny)

ui <- fluidPage(

 titlePanel("Title"),

 sidebarLayout(

  sidebarPanel(

   # Inputs

  ),

  mainPanel(

   # Outputs
```

```
  )
 )
)
server <- function(input, output) {
 # Server logic
}
shinyApp(ui = ui, server = server)
```

```
  - **Inputs**: `sliderInput()`, `selectInput()`, `textInput()`
  - **Outputs**: `plotOutput()`, `tableOutput()`, `textOutput()`

- **Leaflet for Maps**:
  - **Basic Map**:
```

```
library(leaflet)
leaflet() %>%
 addTiles() %>%
 addMarkers(lng=174.768, lat=-36.852, popup="Location")
```

```
  - **Custom Tiles**: `addProviderTiles("Stamen.Toner")`
  - **Polygons**: `addPolygons(data=map_data)`

- **Plotly for Interactive Plots**:
  - **Basic Plot**:
```

```
library(plotly)
plot_ly(data, x=~x, y=~y, type="scatter", mode="markers")
```

By Ahmed Baheeg Khorshid

ver 1.0