『この１冊ですべてわかる　データサイエンスの基本』に収録されている【プログラムコード】

【第 3 章】

＜箱ひげ図　R＞　P57

【Code 1】

```
boxplot(data=InsectSprays, count ~ spray)
```

＜箱ひげ図　Python＞ P58

【Code 2】

```
!pip install pydataset
from pydataset import data
import seaborn as sns

insect = data("InsectSprays")
sns.boxplot(x="spray", y="count", data=insect)
```

＜対応分析　R＞ P61

【Code 1】

```
install.packages("ca")
library(ca)
dat <- apply(HairEyeColor, c(1,2), sum)
dat.ca <- ca(dat)
plot(dat.ca)
legend("bottomright",
       legend=c("Hair","Eye"),
       pch=c(16,17),
       col=c("blue","red"))
```

＜対応分析　Python＞ P63

【Code 2】

```
!pip install pydataset
from pydataset import data
import pandas as pd
import matplotlib.pyplot as plt
!pip install mca
import mca

hec = data("HairEyeColor")
dat = pd.crosstab(index=hec["Hair"],
                  columns=hec["Eye"],
                  values=hec["Freq"],
                  aggfunc="sum")
dat_ca = mca.MCA(dat, benzecri=False)
rows = dat_ca.fs_r(N=2)
cols = dat_ca.fs_c(N=2)

fig, ax=plt.subplots(figsize=(6,6))
ax.scatter(rows[:,0], rows[:,1], c='blue', marker='o', s=50)
labels = dat.index.values
for label,x,y in zip(labels,rows[:,0],rows[:,1]):
    ax.annotate(label, c="blue", xy = (x, y), fontsize=15)

ax.scatter(cols[:,0], cols[:,1], c='red', marker='^', s=50)
labels = dat.columns.values
for label,x,y in zip(labels,cols[:,0],cols[:,1]):
    ax.annotate(label, c="red", xy = (x, y), fontsize=15)

ax.legend(["Hair","Eye"], loc="lower right")
ax.axhline(0, color="black", linestyle="dotted")
ax.axvline(0, color="black", linestyle="dotted")
plt.show()
```

【第 4 章】

＜主成分分析　R＞　P88

【Code 1 】

```
#下準備
library(tidyverse)

#ファイルの読み込み
data <- read_csv("test.csv",
                 locale=locale(encoding = "utf-8"))
#データの概要を示す
glimpse(data)

#主成分分析
res<-prcomp(data[,2:6])
```

【Code 2 】

```
#結果の確認
res
#寄与度など
summary(res)

#バイプロット
biplot(res)
```

【第 5 章】

＜スクレイピング　Python＞　P106、P107

【Code 1】

```
!pip install bs4 # Beautiful Soup のインストール
from urllib import request #urllib.request をインポート
from bs4 import BeautifulSoup # BeautifulSoup をインポート
url = https://www.ds.shiga -u.ac.jp/about/ds/aiming/    #対象の URL
response = request.urlopen(url)
soup = BeautifulSoup(response)
response.close()
print(soup)
```

【Code 2】

```
title = soup.find("h2",class_ = "c-heading--lv2") # title の抽出
print("title :" ,title.text) # .text でテキスト部分のみを抽出する。
sub_title = soup.find_all("h3",class_ = "c-heading--lv3") # sub_title の抽出
content = soup.find_all("p",class_ = "c-txt") # content の抽出
for i in range(len(sub_title)):
    print("sub_title : ", sub_title[i].text)
    print("content : ", content[i].text)
```

＜決定木　Python＞　P115、P116、P117

【Code 1】

```
import pandas as pd
# 可視化で日本語を表示する設定
!pip install japanize-matplotlib
import matplotlib.pyplot as plt
import japanize_matplotlib
import seaborn as sns
sns.set(font="IPAexGothic")
# 決定木を使用するためのライブラリ
from sklearn.tree import DecisionTreeClassifier
# 決定木を可視化するライブラリ
from sklearn.tree import plot_tree
```

```
df = pd.read_csv("/content/drive/MyDrive/sample.csv")
```

【Code 2】

```
X = df.drop(columns = [ 購入])
y = df[ 購入]

# 決定木の深さを指定
depth = 1
# 決定木モデルの作成
tree = DecisionTreeClassifier(max_depth = depth, random_state=0).fit(X, y)

# 決定木の可視化
fig = plt.figure(figsize=(8,8))
plot_tree(tree, feature_names=X.columns, class_names=True, filled=True)
plt.show()
```

【第 6 章】

＜重回帰分析　Python＞　P132、P133、P134

【Code 1 】

```
import pandas as pd
from sklearn import datasets

#データセットの読み込み
diabetes = datasets.load_diabetes()
```

【Code 2 】

```
#データセット情報の表示
print(diabetes[DESCR])
```

【Code 3 】

```
###データセットの表示

#データをデータフレームの形にする
df = pd.DataFrame(diabetes.data)

#列名を指定
df.columns = diabetes.feature_names

#目的変数であるデータをデータフレームに追加
df[Target] = pd.DataFrame(diabetes.target)

#データフレームの大きさを表示
print(df.shape)

#データフレーム先頭 5 行の表示
df.head()
```

【Code 4】

```
#説明変数の作成
X = df[[age, bmi, bp, s1, s2, s3, s4, s5, s6]]

#目的変数の作成
y = df[Target]
```

【Code 5】

```
import statsmodels.api as sm

#回帰モデルを定義
model = sm.OLS(y, sm.add_constant(X))

#モデルの作成
results = model.fit()

#結果の表示
print(results.summary())
```

【第 7 章】

＜ロジスティック回帰分析　R＞　P158

【Code 1】

```
# データの読み込み
data <- read.csv("demo.csv")
# データの先頭 5 人分を表示
head(data,5)
# ロジスティック回帰を実行
resl <- glm(pass~months, data, family="binomial")
# 結果の表示
summary(resl)
```

【第 8 章】

＜区間推定　Python＞　P198

【Code1】

```
import pandas as pd
import numpy as np
from scipy import stats

#　データの読み込み
data = pd.read_csv("./hyo1_data.csv")
#　一列目が ID,二列目が Time
arrive_time = data.iloc[:,1]

#　標本平均
mean = np.mean(arrive_time)
#　不偏分散
var = stats.tvar(arrive_time)
#　データの大きさ
n = 15
print(mean,var,n)
print(stats.norm.interval(confidence=0.95, loc =mean, scale=np.sqrt(var/n)))
```

【第 9 章】

＜統計的仮説検定　R＞　P229、P230

【Code1】

```
# データの読み込み
dat <- read.csv("tea_test.csv")

# A 案と B 案の評価点を取り出す
dat_A <- dat$評価点[dat$評価案 == "A"]
dat_B <- dat$評価点[dat$評価案 == "B"]

# 標本サイズ
n_A <- length(dat_A)
n_B <- length(dat_B)

# 標本平均
mean_A <- mean(dat_A)
mean_B <- mean(dat_B)

# 不偏分散
var_A <- var(dat_A)
var_B <- var(dat_B)

# 2 群の平均値の差の検定
r1 <- qnorm(p = 0.025, mean = 0, sd = sqrt(var_A/n_A + var_B/n_B))
r2 <- qnorm(p = 0.975, mean = 0, sd = sqrt(var_A/n_A + var_B/n_B))
print(c(r1, r2))

# P 値の計算
q <- pnorm(q = abs(mean_A-mean_B), mean = 0, sd = sqrt(var_A/n_A + var_B/n_B))
p <- 2*(1-q)
print(p)
```