

ຮູບບັນແນນະນຳປະໂເກກີ້ພໍາ ກີ່ເໜມາະສມກັບຮ່າງກາຍ ແລະ ຄວາມຊອບຂອງແຕ່ລະບຸຄຄລ

DATA MINING AND ITS APPLICATIONS 05406102

ກລຸ່ມທີ 5

Painpoint

นักศึกษาหลายคนยังเลือกที่พำนัชไม่มีเหตุผล
ชัดเจน

- อย่างการเลิกเล่นกลางคืนเพื่อจะไม่ตรงกับ
ความชอบ
- ไม่แน่ใจว่าที่พำนัชประเภทใดเหมาะสมกับ
ร่างกายและไลฟ์สไตล์ของตนเอง

Motivation

มีคนอยากออกกำลังกาย แต่ไม่รู้ว่าที่พำนัชแบบ
ไหนเหมาะสมกับตัวเอง
พวกเรามีจึงอยากสร้างระบบที่ช่วยให้
นักศึกษาเลือกที่พำนัชได้ตรงกับความชอบและ
พฤติกรรมจึงเป็นโอกาสดีในการนำ Data
Mining มาทำงานกับข้อมูลที่พวกเรามีอยู่

Scope

ข้อมูลที่เราเก็บจากผู้ใช้จริง ภายใน Kmitl เช่นความชอบกีฬา, ส่วนสูง, น้ำหนัก, เพศ, สุขภาพ, พฤติกรรมการออกกำลังกาย

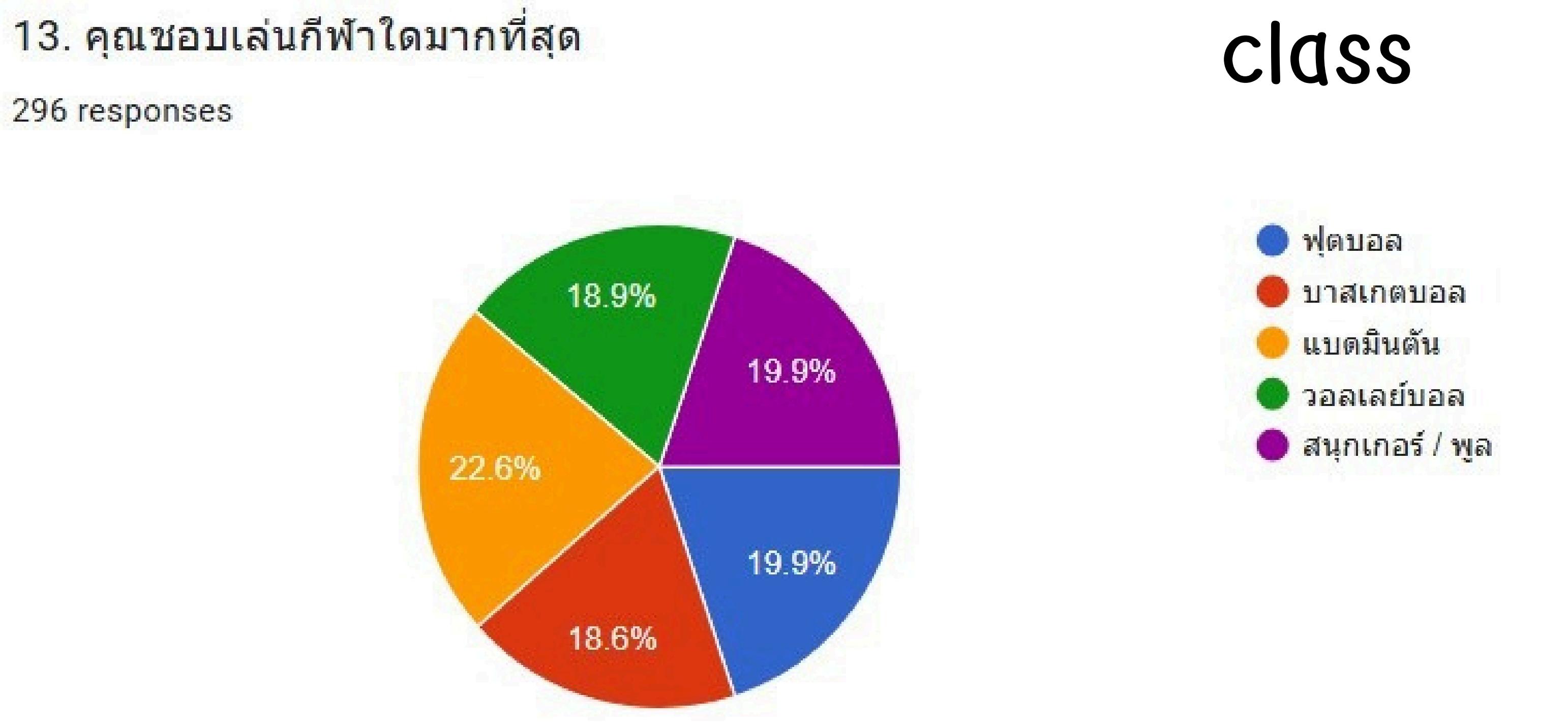


Benefits of the project

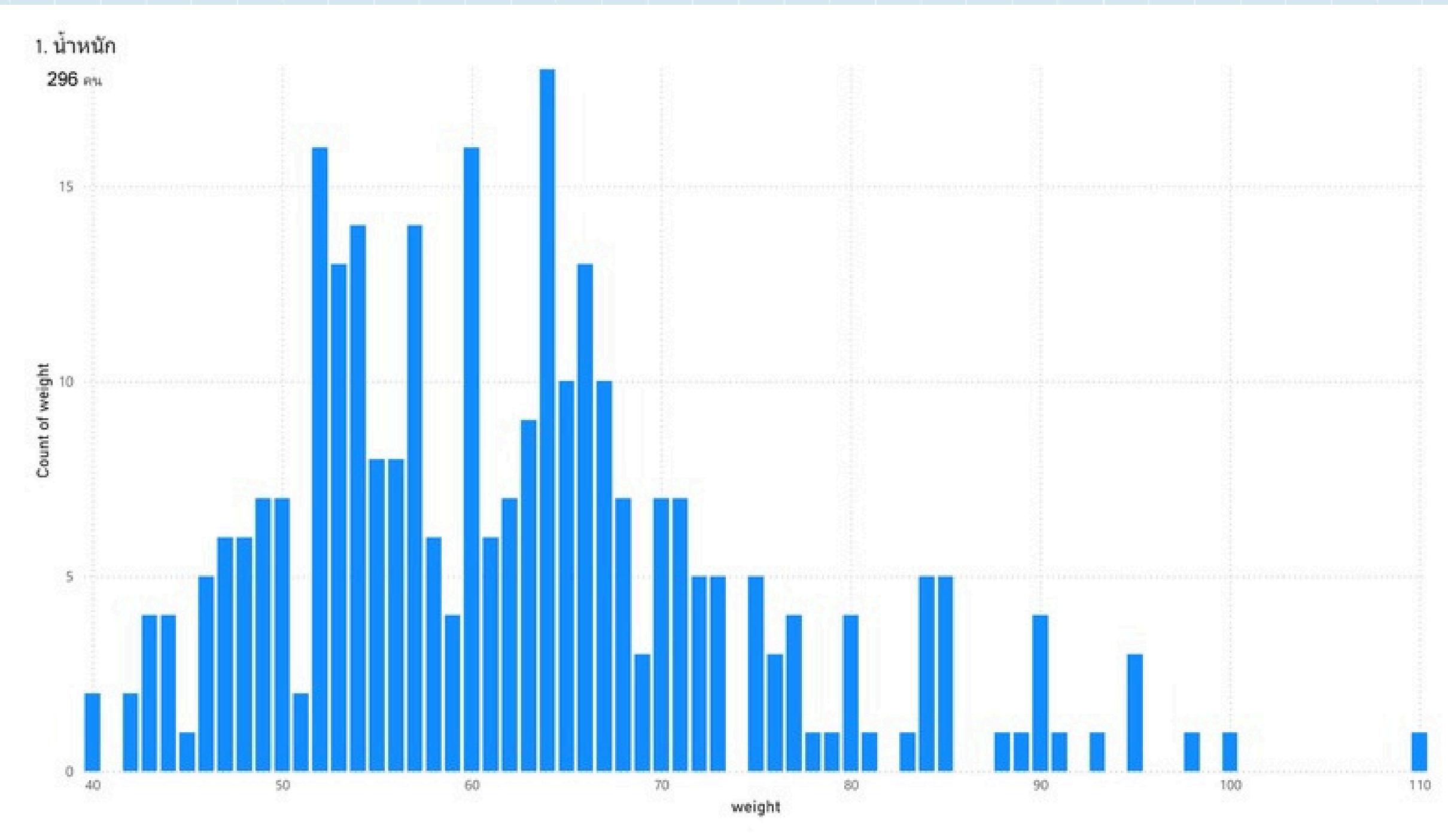
- ผู้ใช้ได้กีฬากีฬาที่เหมาะสมกับตนเอง สุขภาพดีขึ้น
- กระตุ้นให้อยากออกกำลังกาย
- ช่วยให้นักศึกษาที่ไม่รู้จะเล่นกีฬาอะไร หรือกำลังมองหาชั้นเรียน สามารถเลือกกีฬากีฬาที่เหมาะสมกับตนเองได้ง่ายขึ้น
- กลุ่มได้ฝึกทำ Data Mining กับข้อมูลจริง
- สามารถต่อยอดเป็นระบบหรือแอปในอนาคต

Data understanding

จากแบบสอบถามเรามีกําหนด 12 attribute 1 classification(5class)
มีผู้ตอบแบบสอบถามผ่าน google form กําหนด 296 คน



Data understanding



สถิติเบื้องพื้นฐานของน้ำหนัก:

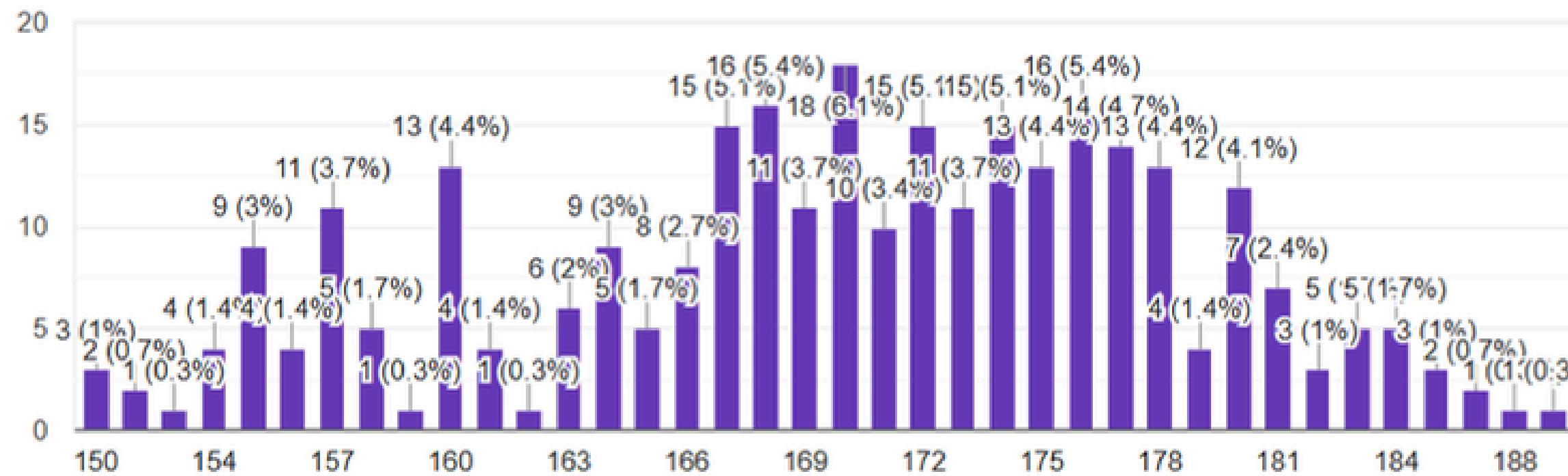
count	293.000000
mean	62.484642
std	12.496737
min	40.000000
25%	53.000000
50%	61.000000
75%	68.000000
max	110.000000

Name: weight, dtype: float64

Data understanding

2. ส่วนสูง

296 responses

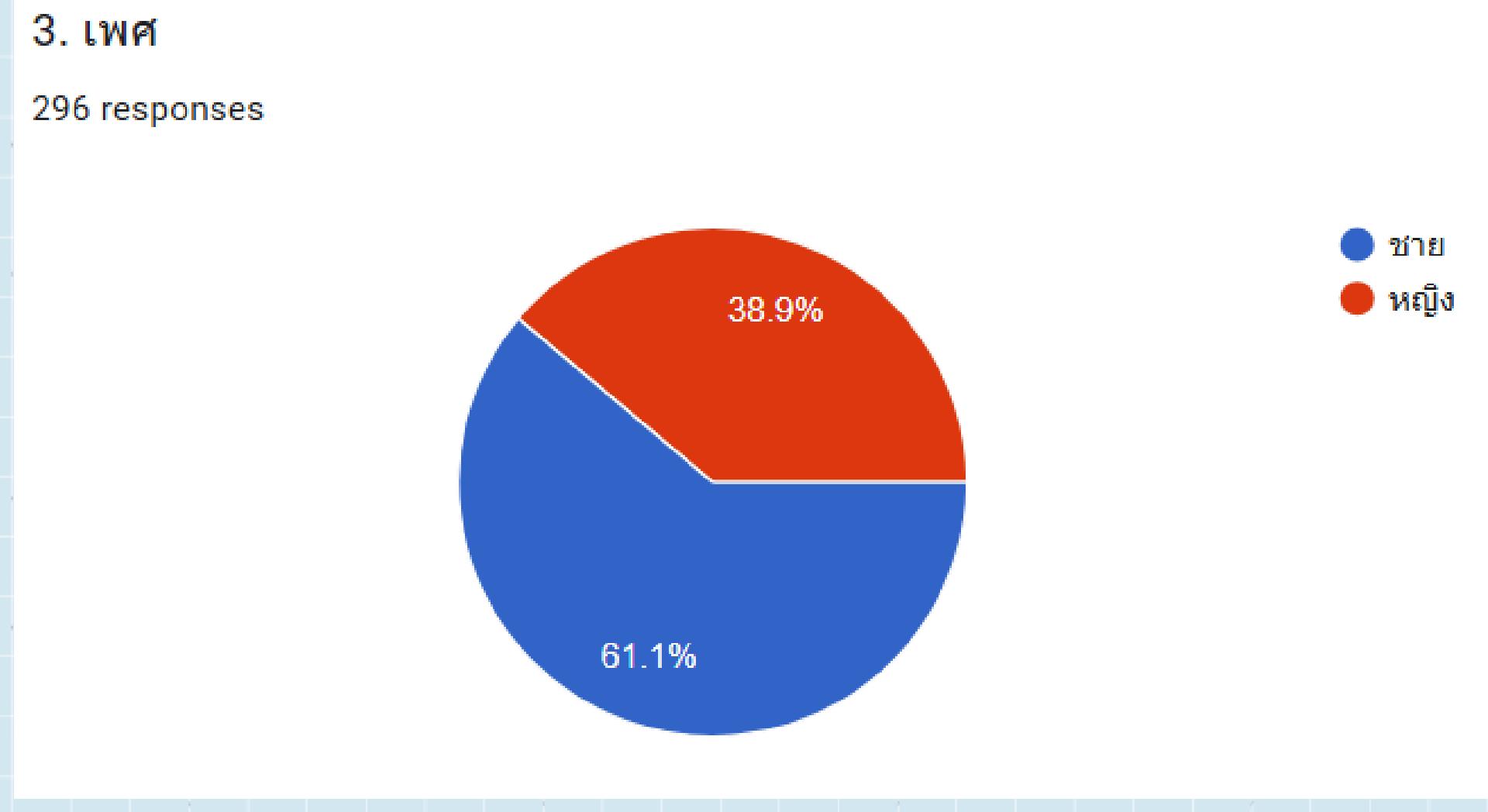


สถิติเชิงพรรณนาของส่วนสูง:

count 293.000000
mean 170.017065
std 8.358597
min 150.000000
25% 165.000000
50% 171.000000
75% 176.000000
max 190.000000

Name: height, dtype: float64

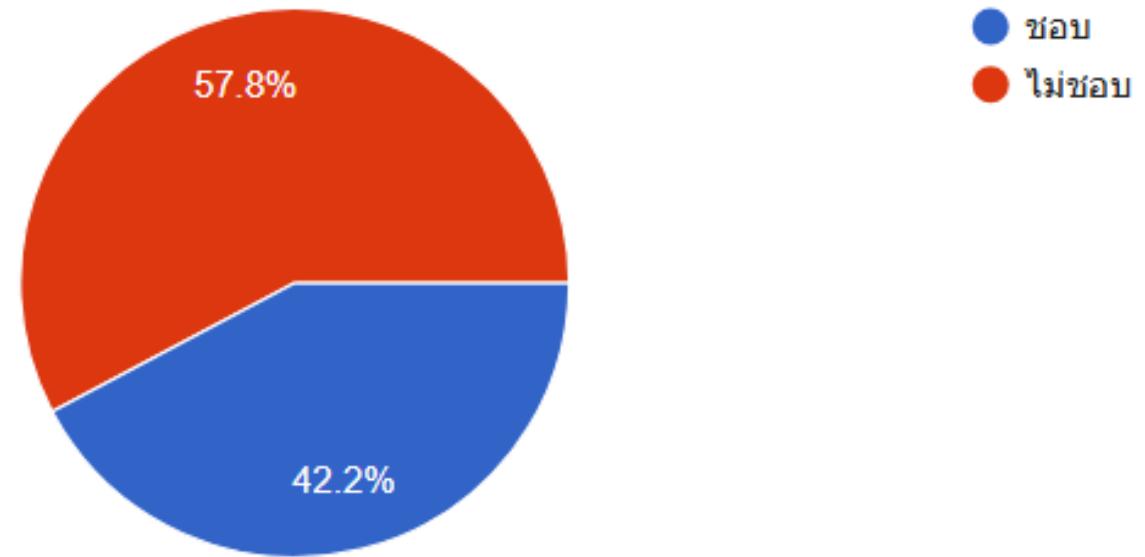
Data understanding



Data understanding

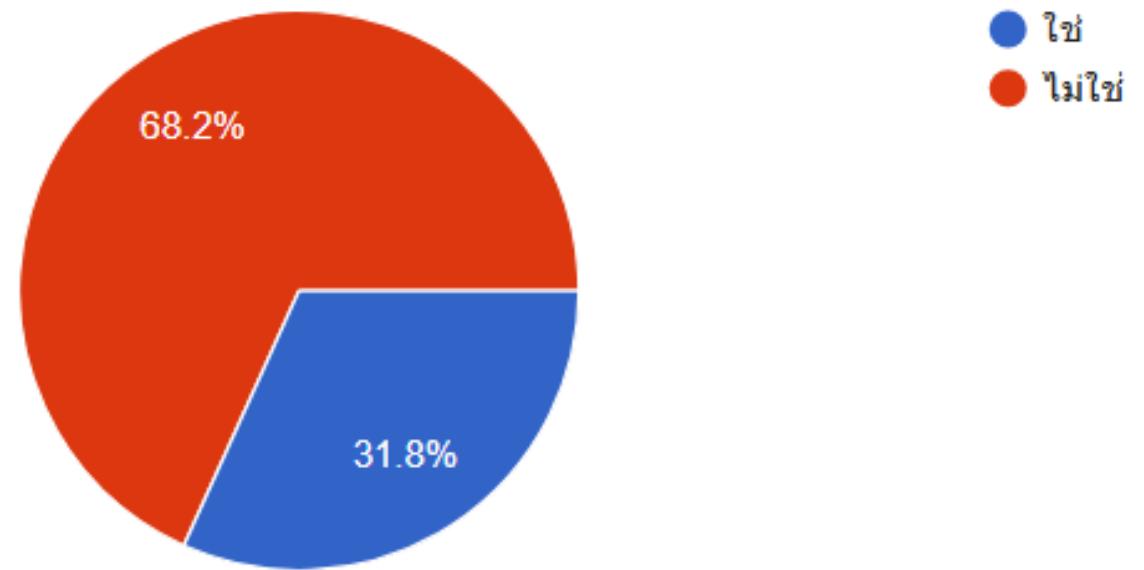
5. คุณชอบกิจกรรมที่มีการชน/ปะทะกับคู่ต่อสู้หรือไม่?

296 responses



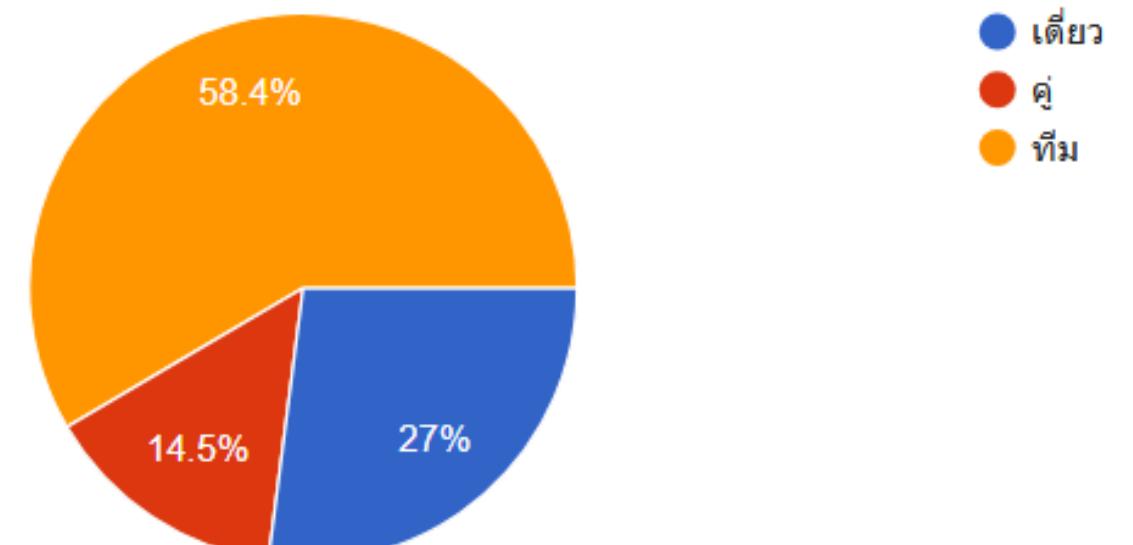
7. คุณเคยมีอาการแน่นหน้าอ ก หอบ หรือเจ็บหน้าอกเวลาทำกิจกรรมทางกายหรือไม่?

296 responses



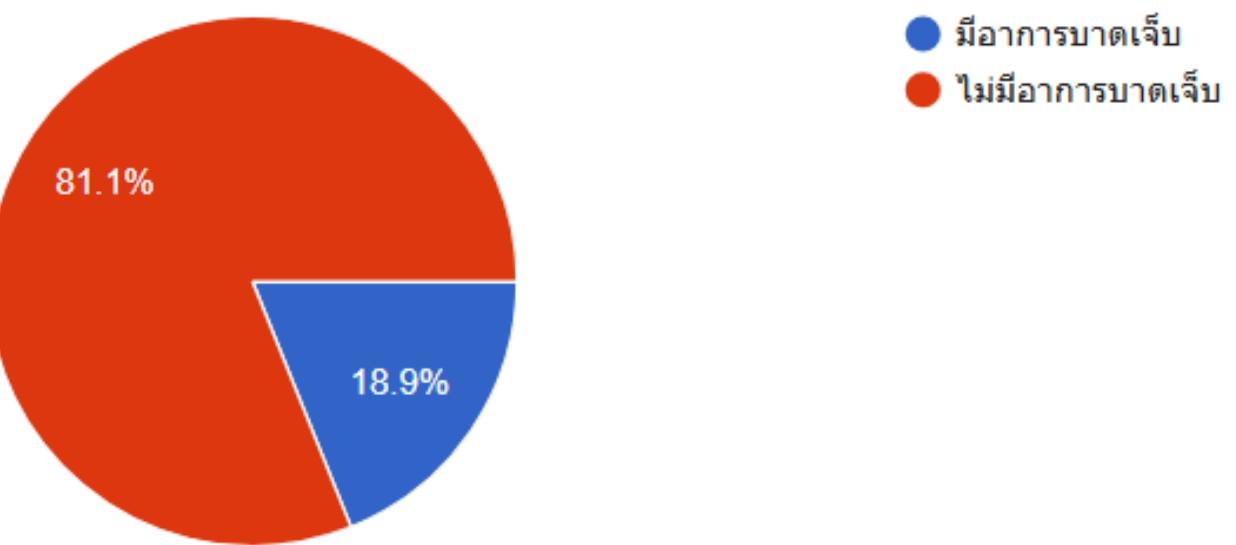
6. คุณชอบเล่นกีฬาแบบเดี่ยว คู่ หรือทีมมากที่สุด?

296 responses



8. ปัจจุบันมีอาการบาดเจ็บที่ข้อเท้าหรือข้อเข่าที่กราฟบนต่อการเล่นกีฬาหรือไม่?

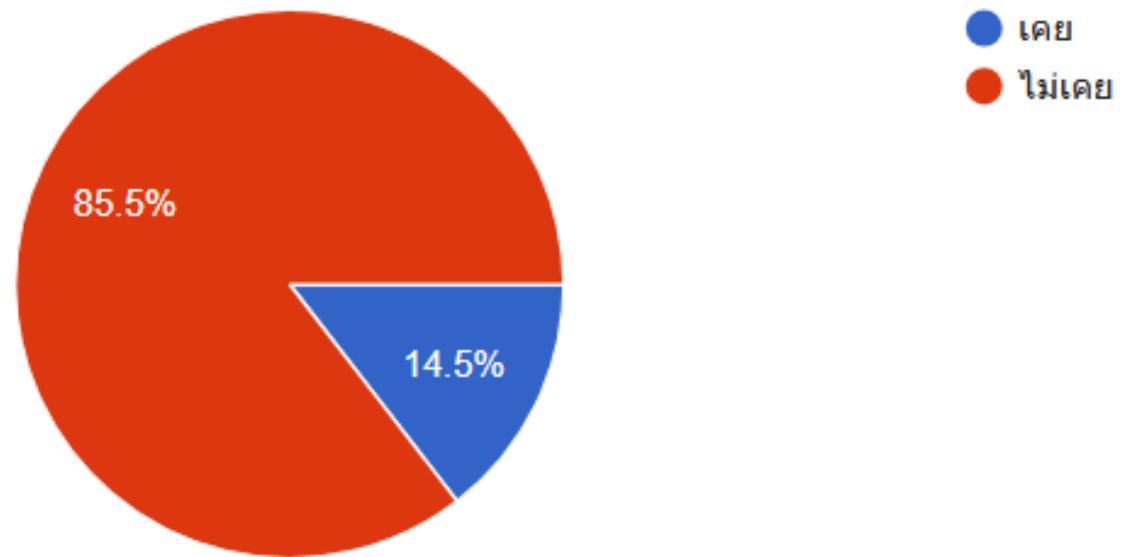
296 responses



Data understanding

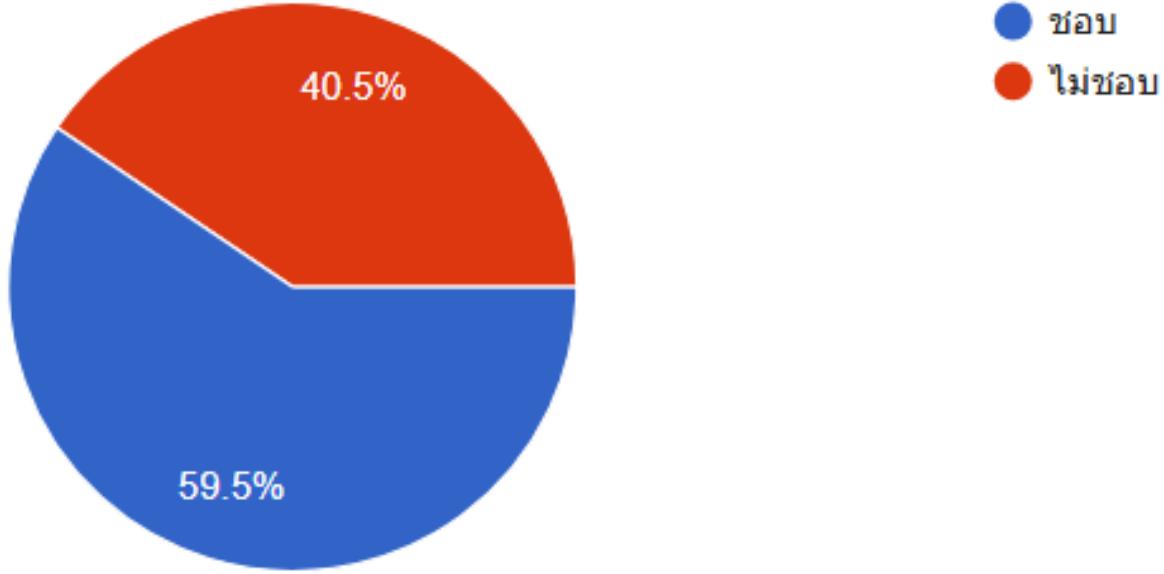
9. คุณเคยมีประวัติเป็นโรคหัวใจหรือโรคปอด (เช่น หอบหืด, ปอดอุดกั้นเรื้อรัง) หรือไม่?

296 responses



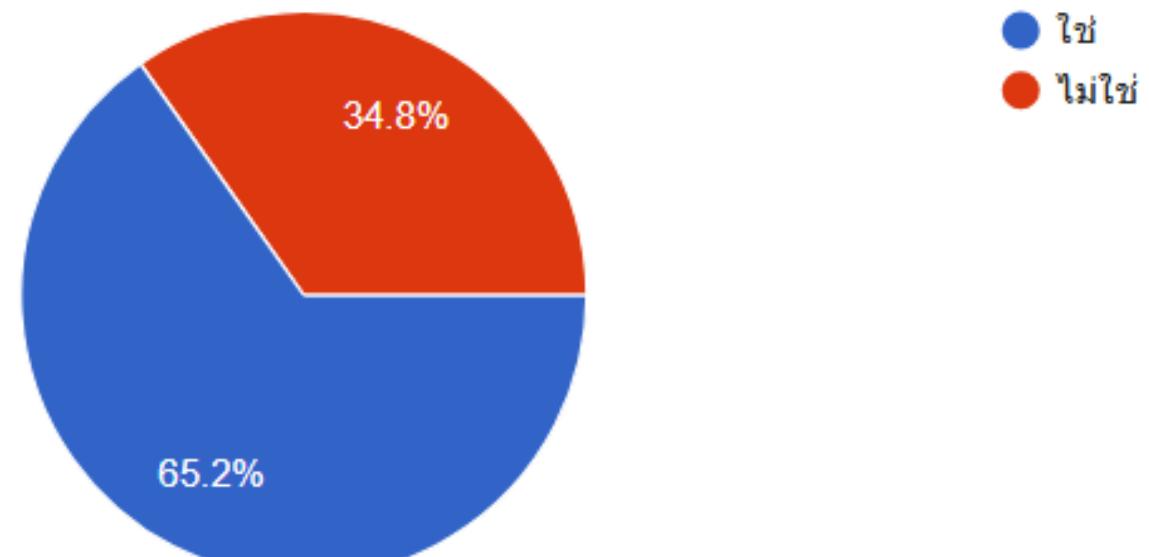
10. คุณชอบเล่นกีฬาที่ต้องการโดดหรือเคลื่อนตัวขึ้นสูงหรือไม่?

296 responses



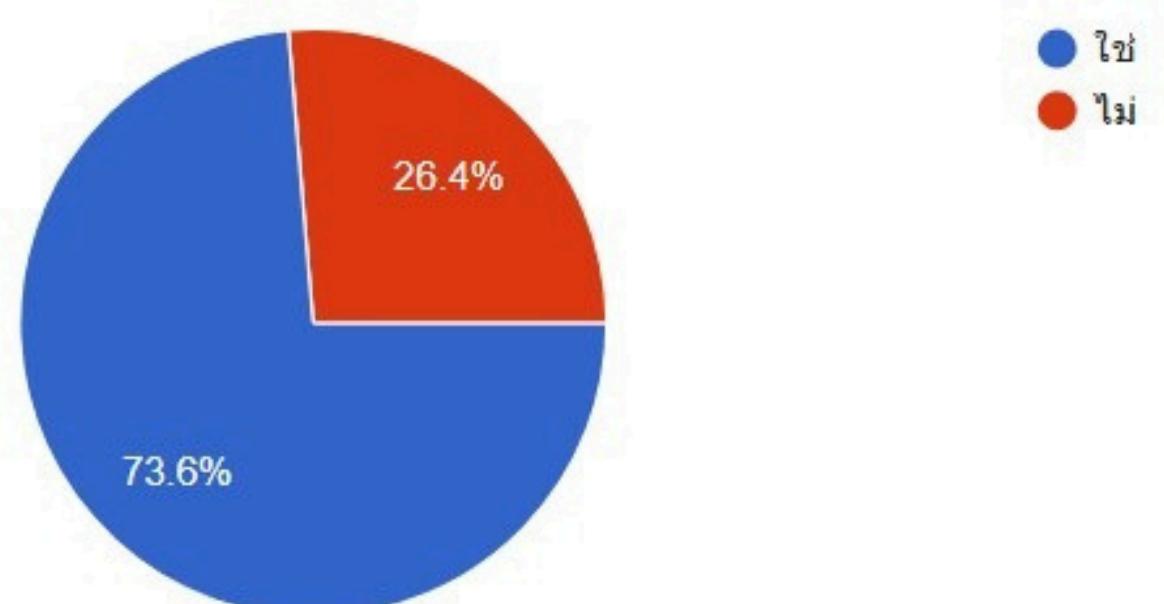
11. คุณชอบกิจกรรมที่ต้องใช้ความอดทนและใช้เวลานานหรือไม่?

296 responses



12. คุณชอบกิจกรรมที่ต้องใช้ความคล่องตัวและการเคลื่อนไหวรวดเร็วหรือไม่?

296 responses



Data Preprocessing



แปลงข้อมูลภาษาไทยเป็นอังกฤษ เพื่อให้วิเคราะห์ใน google colab ได้ง่าย

A	B	C	D	E	F	G	H	I	J	K	L	M	N
weight	height	gender	location	Physical_Condition	play_type	chest_symptom	injury	disease	jumping	endurance	agility	preferred_sport	
64	172	male	indoor	dislike	pair	no	yes	no	no	no	no	snooker or pool	
57	168	female	indoor	dislike	team	no	no	no	yes	yes	yes	volleyball	
65	160	female	indoor	dislike	solo	no	yes	no	yes	no	no	badminton	
53	170	female	indoor	like	team	no	no	no	yes	yes	yes	badminton	
57	177	male	outdoor	like	solo	no	no	no	yes	yes	yes	snooker or pool	
56	163	female	indoor	dislike	team	yes	no	no	yes	no	yes	badminton	
55	174	male	outdoor	like	team	no	no	no	yes	yes	yes	football	
66	185	male	indoor	like	solo	no	no	no	yes	yes	yes	badminton	
47	155	female	indoor	dislike	solo	yes	no	no	no	no	yes	badminton	
85	178	male	outdoor	like	team	yes	no	yes	yes	yes	yes	basketball	
78	170	male	outdoor	like	team	no	yes	no	no	no	yes	football	
65	176	male	outdoor	like	team	yes	no	no	yes	yes	yes	basketball	
60	160	female	indoor	like	pair	no	no	no	no	no	no	snooker or pool	
58	163	female	outdoor	dislike	team	no	no	no	yes	yes	yes	basketball	
46	158	female	indoor	dislike	solo	no	no	no	yes	yes	yes	badminton	
42	152	female	indoor	dislike	solo	no	no	no	yes	yes	yes	badminton	
52	161	female	indoor	like	team	yes	no	no	yes	yes	yes	volleyball	
49	153	female	indoor	dislike	pair	yes	no	no	no	no	no	badminton	
52	175	female	outdoor	like	team	no	no	no	yes	yes	yes	basketball	
44	150	female	indoor	dislike	pair	yes	yes	no	no	no	no	badminton	
50	157	female	indoor	dislike	solo	no	no	no	no	yes	no	badminton	
91	170	male	indoor	like	solo	no	no	no	yes	no	yes	badminton	
83	176	male	indoor	dislike	team	yes	yes	no	yes	no	yes	volleyball	
77	177	male	indoor	dislike	team	no	no	no	yes	yes	yes	volleyball	
66	164	female	indoor	dislike	solo	yes	no	no	no	yes	yes	badminton	

Data Cleaning

1. ตรวจสอบ missing และ duplicates

```
▶ print("Missing values per column:")    ##### เช็คค่าว่าง  
print(df.isnull().sum())  
  
print("\nNumber of duplicate rows:")    ##### เช็คค่าซ้ำ  
print(df.duplicated().sum())
```

→ Missing values per column:

weight	0
height	0
gender	0
location	0
Physical Contact	0
play_type	0
chest_symptom	0
injury	0
disease	0
jumping	0
endurance	0
agility	0
preferred_sport	0
dtype: int64	

Number of duplicate rows:

3

```
import pandas as pd  
df = pd.read_csv("/content/english_predict_sport.csv")  
print(df.shape)
```

→ (296, 13)

	weight	height	gender	location	Physical Contact	play_type	chest_symptom	injury	disease	jumping	endurance	agility	preferred_sport
94	66	172	male	outdoor	like	team		no	no	yes	yes	yes	football
98	52	174	male	outdoor	like	team		no	no	no	yes	yes	football
150	75	170	male	indoor	dislike	pair		yes	no	no	no	no	badminton
248	66	172	male	outdoor	like	team		no	no	yes	yes	yes	football
249	75	170	male	indoor	dislike	pair		yes	no	no	no	yes	badminton
263	52	174	male	outdoor	like	team		no	no	no	yes	yes	football

-ไม่มีค่า missing
-WUแกรนท์ซ้ำกัน 3 ครั้ง

3

Data Cleaning

2. กำจัดค่า duplicates

```
df.drop_duplicates(inplace=True)

print("Number of duplicate rows after removal:")
print(df.duplicated().sum())

print("\nShape of the dataframe after removing duplicates:")
print(df.shape)
```

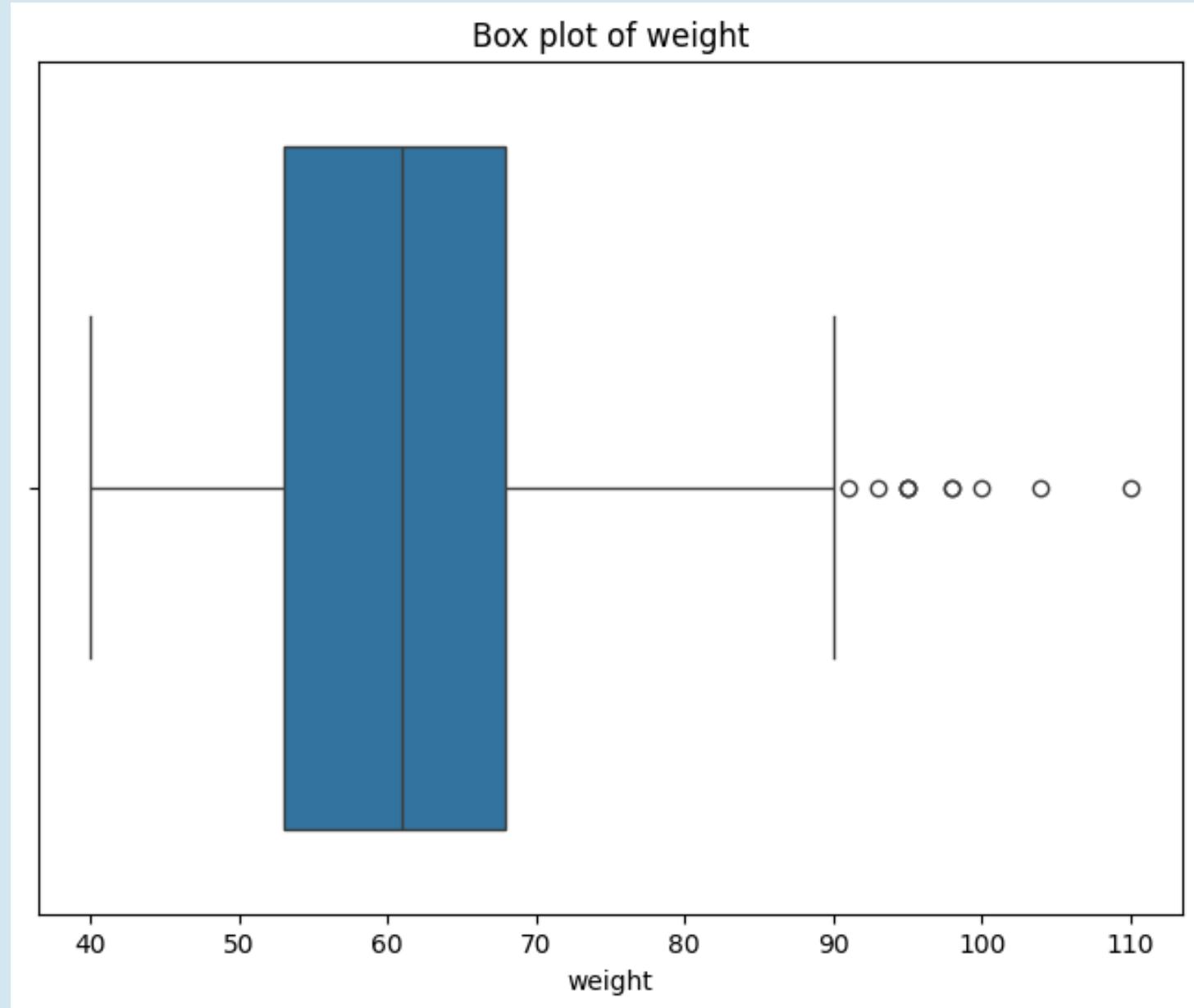
Number of duplicate rows after removal:

0

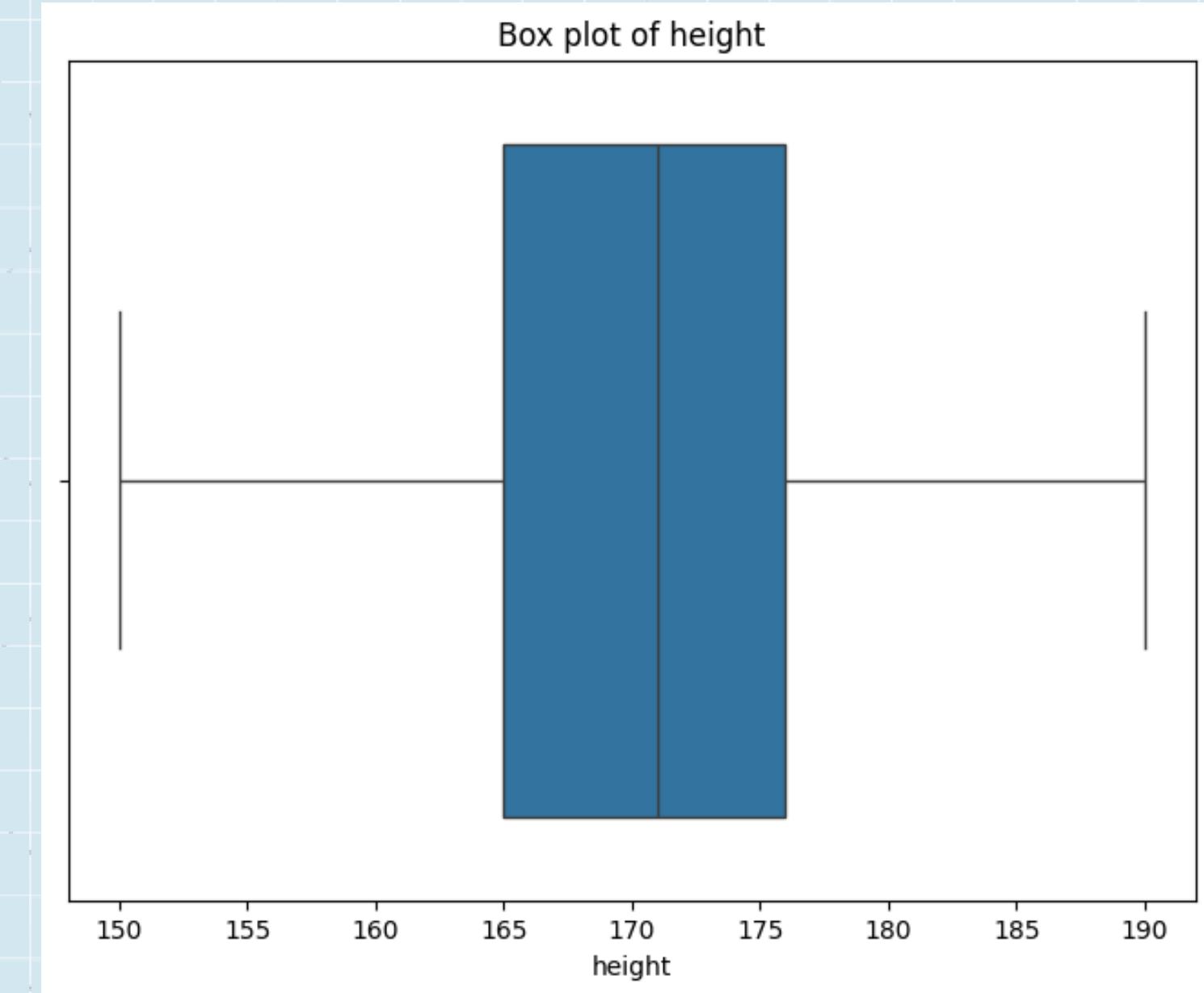
Shape of the dataframe after removing duplicates:

(293, 13)

Outlier Analysis



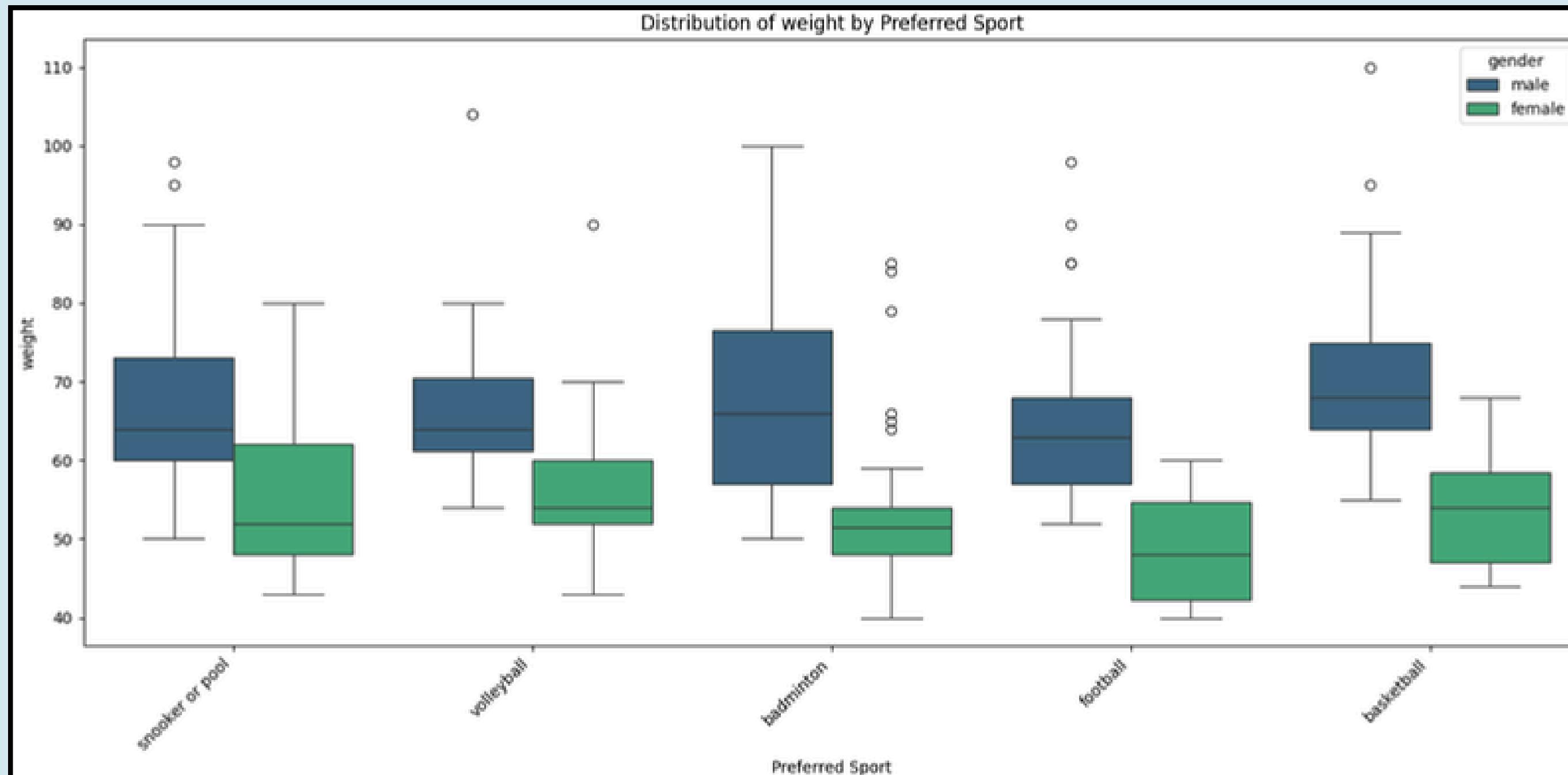
- นำหนัก เบ๊ขวา และ มี outlier ค่อนข้างเยอะ



- ส่วนสูงข้อมูลมีการแจกแจงปกติ ไม่พบ outlier

Outlier Analysis

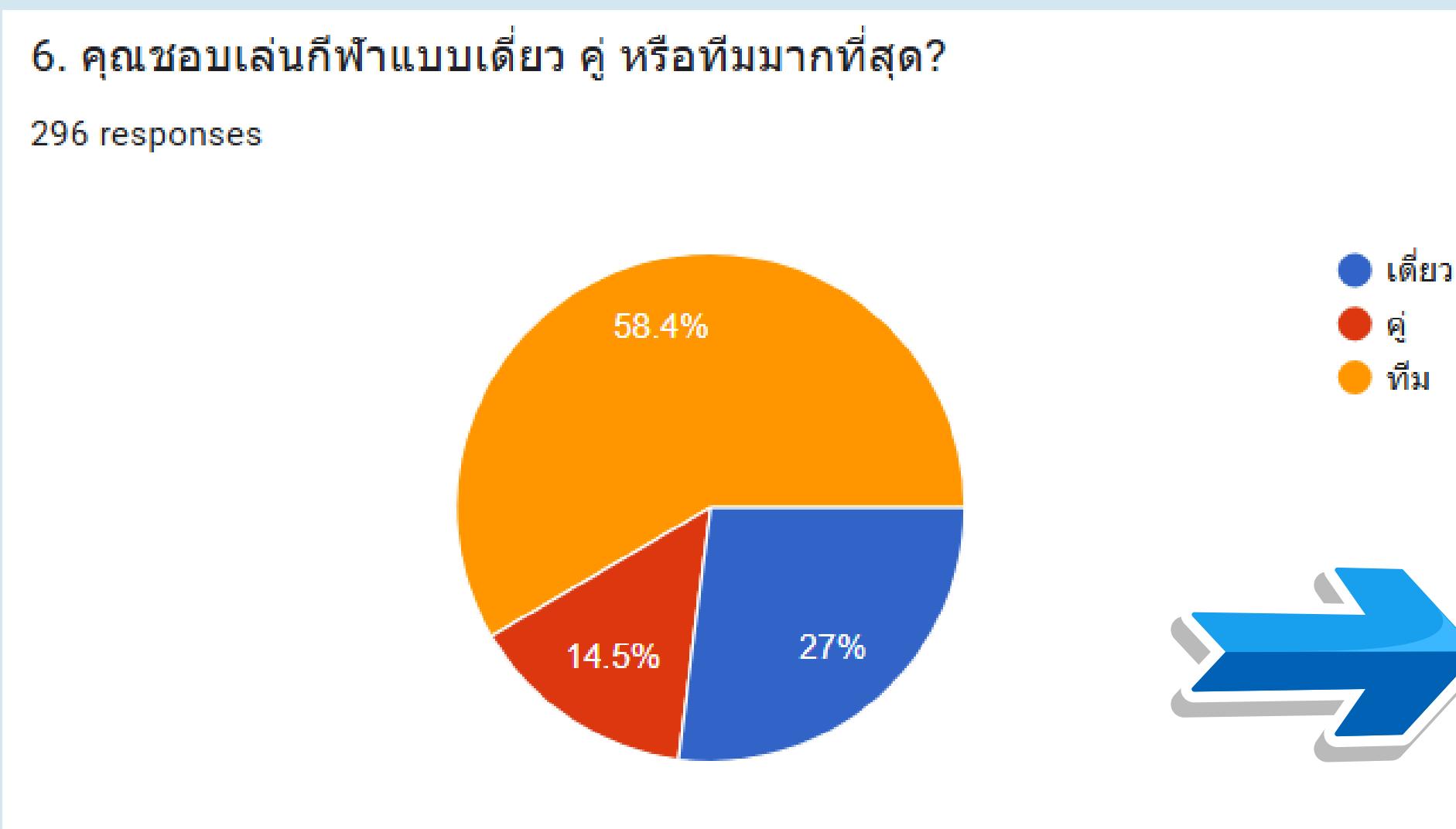
boxplot หาความสัมพันธ์ ระหว่างน้ำหนัก โดยแยกตามกีฬาและเพศ



จาก Weight ก็มี outlier ค่อนข้างเยอะ เราจึงพิจารณาดูความสัมพันธ์ กับกีฬาและเพศ pragmatism ที่สัมพันธ์กับน้ำหนักชัดเจน แต่เมื่อแยกตามกีฬาไม่พบ pattern ก็ชัดเจน จึงทำการ ตัดฟีเจอร์ Weight ออก

Transformation

1. รวมค่า Solo & Pair ในฟีเจอร์ Play Type → Solo_Pair



```
df['playtype'] = df['play_type'].replace(['solo', 'pair'], 'solo_pair')
display(df[['play_type', 'playtype']].head())
df = df.drop(columns=['play_type'])
```

	play_type	playtype	
0	pair	solo_pair	
1	team	team	
2	solo	solo_pair	
3	team	team	
4	solo	solo_pair	

ฟีเจอร์ Play Type เดิมมีค่า 3 แบบคือ Solo, Pair, Team แต่ตามความเป็นจริง Solo และ Pair มีเพียง 2 กีฬา คือสนุ๊กเกอร์กับแบดมินตัน ซึ่งมีลักษณะคล้ายกันคือเล่นได้กึ่งเดี่ยวและคู่ ไม่ใช่ทีมใหญ่ เราจึงรวม Solo และ Pair เป็นกลุ่มเดียวกันกว่า Solo_Pair เพื่อให้ feature นี้สะท้อนประเภทกีฬาชัดเจนขึ้น และลดปัญหาสัดส่วนข้อมูลกีฬาระจายไม่สมดุล

Transformation

ກ່ອນແປລັງເປົ້າ Label Encoding

```
1 df.drop('play_type',axis=1,inplace=True)
2 df
3
```

	height	gender	location	Physical Contact	chest_symptom	injury	disease	jumping	endurance	agility	preferred_sport	playtype
0	172	male	indoor		dislike	no	yes	no	no	no	snooker or pool	solo_pair
1	168	female	indoor		dislike	no	no	no	yes	yes	yes	volleyball
2	160	female	indoor		dislike	no	yes	no	yes	no	no	badminton
3	170	female	indoor		like	no	no	no	yes	yes	yes	badminton
4	177	male	outdoor		like	no	no	no	yes	yes	yes	snooker or pool
...
291	179	male	indoor		like	yes	no	no	yes	yes	yes	badminton
292	173	male	indoor		like	yes	yes	no	no	yes	no	snooker or pool
293	176	male	outdoor		like	no	no	no	yes	yes	yes	football
294	176	female	indoor		dislike	no	no	no	yes	yes	yes	volleyball
295	180	male	indoor		like	yes	no	no	yes	no	no	volleyball

293 rows × 12 columns

ບໍລິຫານນີ້ແມ່ນ:[ສໍາເລັດສ່າຍ df](#) | [New interactive sheet](#)

Transformation

หลังแปลงเป็น Label Encoding

```
1 #Label Encoding
2 from sklearn.preprocessing import LabelEncoder
3 le = LabelEncoder()

1 df.columns
2 Index(['height', 'gender', 'location', 'Physical Contact', 'chest_symptom',
   'injury', 'disease', 'jumping', 'endurance', 'agility',
   'preferred_sport', 'playtype'],
   dtype='object')

1 col_to_le = [ 'gender', 'location', 'Physical Contact',
   'chest_symptom', 'injury', 'disease', 'jumping', 'endurance', 'agility',
   'playtype']
2 for col in col_to_le:
3     df[col] = le.fit_transform(df[col])
4 df['preferred_sport'] = le.fit_transform(df['preferred_sport'])+1
5 df

height  gender  location  Physical Contact  chest_symptom  injury  disease  jumping  endurance  agility  preferred_sport  playtype
0      172       1         0                  0            0       1       0       0       0       0       0       4       0
1      168       0         0                  0            0       0       0       1       1       1       1       5       1
2      160       0         0                  0            0       1       0       1       0       0       0       1       0
3      170       0         0                  1            0       0       0       1       1       1       1       1       1
4      177       1         1                  1            0       0       0       1       1       1       1       4       0
...
291    179       1         0                  1            1       0       0       1       1       1       1       1       1
292    173       1         0                  1            1       1       0       0       1       0       4       0
293    176       1         1                  1            0       0       0       1       1       1       3       0
294    176       0         0                  0            0       0       0       1       1       1       5       1
295    180       1         0                  1            1       0       0       1       0       0       5       1

293 rows × 12 columns
```

Dataset Description & Feature Encoding

1	น้ำหนัก	weight	8	ปัจจุบันมีอาการบาดเจ็บหรือไม่เป็นเพื่อการแพทย์ด้านกีฬาใน ไม่มีอาการบาดเจ็บ	injury	
2	ส่วนสูง	height		มีอาการบาดเจ็บ	no	0
3	เพศ	gender	9	เดบิบีฟาร์ฟีฟินโรคฟ้าไจนหรือโรคปอด (เช่น หอบหืด) หรือไม่	disease	
	หญิง	female		ไม่เดบิบ	no	0
	ชาย	male		เดบิบ	yes	1
4	คุณชอบกีฬาชนิดใด หรือ กีฬาใด	location	10	คุณชอบเล่นกีฬาที่ต้องการใช้ความเร็วในการเคลื่อนไหวมากที่สุดใน ไม่ชอบ	jumping	
	ในร่ม	indoor		ชอบ	no	0
	กลางแจ้ง	outdoor		ไม่ชอบ	yes	1
5	คุณชอบกีฬาชนิดใดมากที่สุด/ปานกลางคุณชอบกีฬาใด	Physical Contact	11	คุณชอบกีฬาชนิดใดที่ต้องใช้ความต้องทนและใช้เวลานานที่สุดใน	endurance	
	ไม่ชอบ	dislike		ไม่ใช่	no	0
	ชอบ	like		ใช่	yes	1
6	คุณชอบเล่นกีฬาแบบเดี่ยว คู่ หรือทีมมากที่สุด	playtype	12	คุณชอบกีฬาชนิดใดที่ต้องใช้ความต้องทนและการเคลื่อนไหวไวมากที่สุดใน	agility	
	เดี่ยวและคู่	solo_pair		ไม่ใช่	no	0
	ทีม	team		ใช่	yes	1
7	เคยมีอาการแน่นหน้าอก หอบ หรือเจ็บหน้าอก เวลาทำการกีฬาอย่างมากใน	chest_symptom	13	คุณชอบเล่นกีฬาใดมากที่สุด	preferred_sport	
	ไม่ใช่	no		แบดมินตัน	badminton	1
	ใช่	yes		บาสเกตบอล	basketball	2
				ฟุตบอล	football	3
				สนุกเกอร์ / พูล	snooker or pool	4
				วอลเลย์บอล	volleyball	5

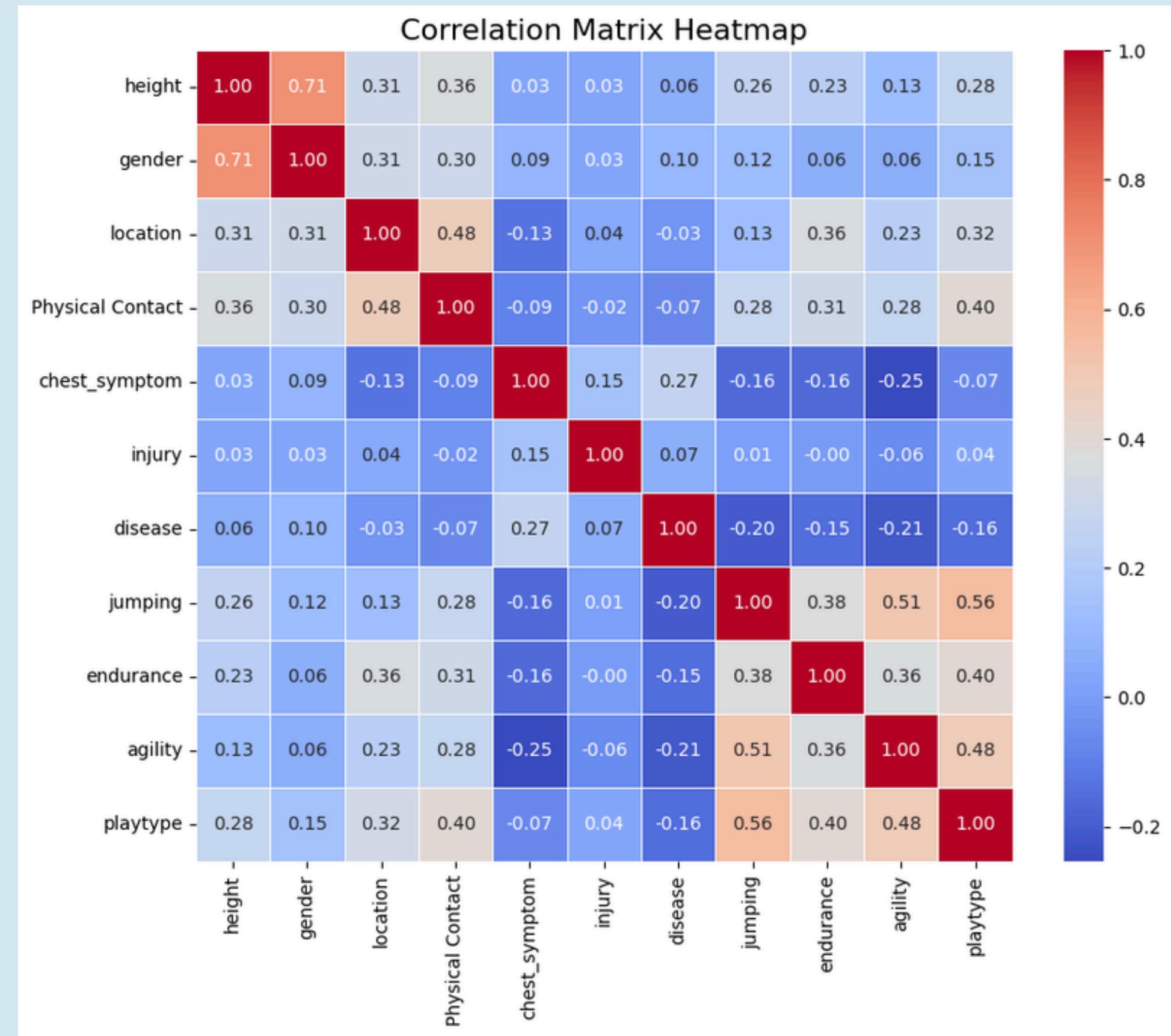
Feature Scaling: Standardization of Height

```
from sklearn.preprocessing import StandardScaler  
scaler = StandardScaler()  
df_final["height"] = scaler.fit_transform(df_final[["height"]])  
display(df_final.head())
```

	height	gender	location
0	0.237639	1	0
1	-0.241729	0	0
2	-1.200465	0	0
3	-0.002045	0	0
4	0.836849	1	1

เลือกใช้ StandardScaler กับฟีเจอร์ Height เนื่องจากเป็นข้อมูลเชิงตัวเลข เพื่อปรับค่าให้อยู่ในสเกลเดียวกัน ทำให้มีค่าเฉลี่ย = 0 และ ส่วนเบี่ยงเบนมาตรฐาน = 1 ชั่งช่วยให้โมเดลเรียนรู้ได้อย่างสมดุล โดยเฉพาะอัลกอริทึมที่ไม่ต้องสเกล เช่น SVM และ KNN

Heatmap តុកវាមសំណើន៍រៀនខ្លះខ្លែងពោត៊ន



Modeling and Performance of the model



Train-Test Split

```
from sklearn.model_selection import train_test_split

# Define features (X) and target (y) from the scaled dataframe
X_scaled = df_final.drop(columns=['preferred_sport_encoded'])
y_scaled = df_final['preferred_sport_encoded']

X_train_scaled, X_test_scaled, y_train_scaled, y_test_scaled = train_test_split(X_scaled, y_scaled, test_size=0.2, random_state=42)

print("Shape of X_train_scaled:", X_train_scaled.shape)
print("Shape of X_test_scaled:", X_test_scaled.shape)
print("Shape of y_train_scaled:", y_train_scaled.shape)
print("Shape of y_test_scaled:", y_test_scaled.shape)
```

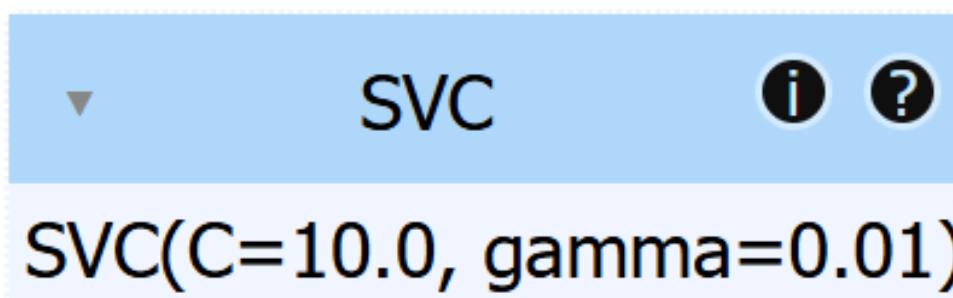
```
Shape of X_train_scaled: (234, 11)
Shape of X_test_scaled: (59, 11)
Shape of y_train_scaled: (234,)
Shape of y_test_scaled: (59,)
```

แบ่งข้อมูลออกเป็น Train 80% = 234 ตัวอย่าง และ Test 20% = 59 ตัวอย่าง เพื่อให้โมเดลได้เรียนรู้จากข้อมูลส่วนใหญ่ และประเมินผลกับข้อมูลที่ไม่เคยเห็นมาก่อน ก่อนเข้าสู่ขั้นตอน Modeling

โมเดลที่นำมาทำการฝึก

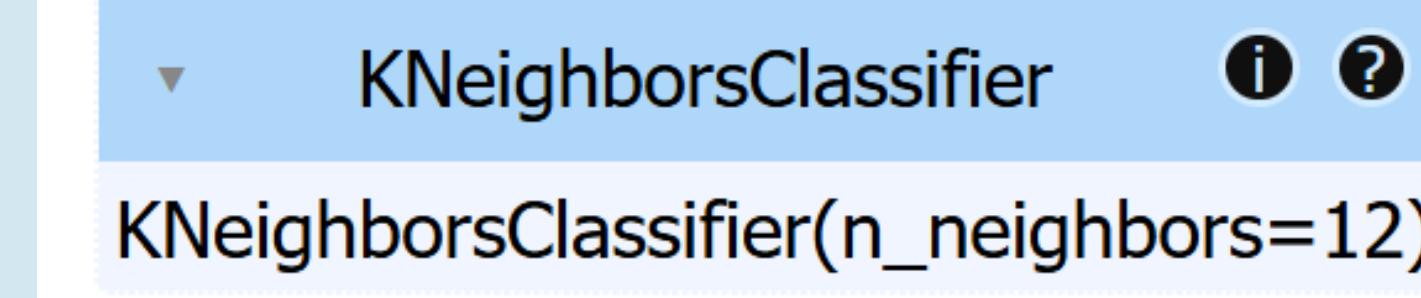
1. Support vector machine

```
from sklearn.svm import SVC  
svm_model = SVC(C=10.0, gamma=0.01,  
)  
svm_model.fit(X_train_scaled, y_train_scaled)
```



2. K-nearest neighbor

```
from sklearn.neighbors import KNeighborsClassifier  
knn_model = KNeighborsClassifier(n_neighbors=12)  
knn_model.fit(X_train_scaled, y_train_scaled)
```



ໂມໂດລກື່ນໍາມາກຳກາຮືຟ

3. Decision Tree

```
from sklearn.tree import DecisionTreeClassifier  
decision_tree_model = DecisionTreeClassifier(  
    max_depth=3,  
    min_samples_leaf=1,  
    ccp_alpha=0.0,  
    random_state=42  
)  
decision_tree_model.fit(X_train_scaled, y_train_scaled)
```

DecisionTreeClassifier



DecisionTreeClassifier(max_depth=3, random_state=42)

4. Random forest

```
from sklearn.ensemble import RandomForestClassifier  
random_forest_model = RandomForestClassifier(  
    n_estimators=240,  
    min_samples_split=2,  
    min_samples_leaf=2,  
    max_features='sqrt',  
    max_depth=10,  
    bootstrap=True,  
    random_state=42,  
)  
random_forest_model.fit(X_train_scaled, y_train_scaled)
```

RandomForestClassifier



RandomForestClassifier(max_depth=10, min_samples_leaf=2, n_estimators=240, random_state=42)

Evaluation : Cross Validation (5-fold)

```
import numpy as np
from sklearn.model_selection import KFold, cross_validate

k_fold = KFold(n_splits=5, shuffle=True, random_state=42)
scoring = { 'accuracy': 'accuracy', 'f1': 'f1_weighted', 'precision': 'precision_weighted','recall': 'recall_weighted'}
models_for_cv = {"SVM": svm_model,"KNN": knn_model,"Decision Tree": decision_tree_model, "Random Forest": random_forest_model,}

print("--- Cross-validation Results (5-fold) train set")
for name, model in models_for_cv.items():
    cv = cross_validate(model, X_train_scaled, y_train_scaled, cv=k_fold, scoring=scoring, n_jobs=-1,
    return_train_score=False )
    acc = cv['test_accuracy']
    f1 = cv['test_f1']
    pre = cv['test_precision']
    rec = cv['test_recall']

    def line(title, arr):
        mean_pct = arr.mean() * 100
        ci95_pct = arr.std() * 200 # 2*std ภายใน ~95% CI แนะนำ
        arr_pct = np.round(arr * 100, 2)
        print(f" {title}: {arr_pct} Mean = {mean_pct:.2f}% (+/- {ci95_pct:.2f}%)")

    print(f"\n{name}")
    line("Accuracy", acc)
    line("F1-score", f1)
    line("Precision", pre)
    line("Recall", rec)
```

Evaluation Model : Result of Cross Validation (5-fold)

--- Cross-validation Results (5-fold) train set

SVM

Accuracy : [74.47 78.72 74.47 78.72 73.91] Mean = 76.06% (+/- 4.37%)
F1-score : [73.77 78.21 74.55 79.66 73.48] Mean = 75.93% (+/- 5.04%)
Precision: [77.07 78.94 77.18 83.78 76.48] Mean = 78.69% (+/- 5.35%)
Recall : [74.47 78.72 74.47 78.72 73.91] Mean = 76.06% (+/- 4.37%)

KNN

Accuracy : [74.47 68.09 76.6 68.09 71.74] Mean = 71.79% (+/- 6.80%)
F1-score : [74.78 68.6 77.37 68.9 71.39] Mean = 72.21% (+/- 6.81%)
Precision: [76.77 72.37 80.06 72.99 71.25] Mean = 74.69% (+/- 6.53%)
Recall : [74.47 68.09 76.6 68.09 71.74] Mean = 71.79% (+/- 6.80%)

Decision Tree

Accuracy : [70.21 70.21 76.6 74.47 69.57] Mean = 72.21% (+/- 5.61%)
F1-score : [71.93 69.4 76.76 75.57 69.93] Mean = 72.72% (+/- 5.93%)
Precision: [74.83 73.78 78.21 77.81 70.93] Mean = 75.11% (+/- 5.38%)
Recall : [70.21 70.21 76.6 74.47 69.57] Mean = 72.21% (+/- 5.61%)

Random Forest

Accuracy : [78.72 74.47 76.6 78.72 76.09] Mean = 76.92% (+/- 3.26%)
F1-score : [78.89 73.45 76.68 79.34 75.16] Mean = 76.71% (+/- 4.45%)
Precision: [79.76 74.56 78.54 82.86 77.53] Mean = 78.65% (+/- 5.43%)
Recall : [78.72 74.47 76.6 78.72 76.09] Mean = 76.92% (+/- 3.26%)

Evaluation Model : 20% Test set

```
from sklearn.metrics import accuracy_score, f1_score, precision_score, recall_score
print("ประเมินโมเดล ของ Test Set: ")
for name, model in models_for_cv.items():
    y_pred = model.predict(X_test_scaled)

    accuracy = accuracy_score(y_test_scaled, y_pred)
    f1      = f1_score(y_test_scaled, y_pred, average='weighted')
    precision = precision_score(y_test_scaled, y_pred, average='weighted')
    recall   = recall_score(y_test_scaled, y_pred, average='weighted')

    print(f"\n{name}")
    print(f" Accuracy : {accuracy:.4f}")
    print(f" F1-score : {f1:.4f}")
    print(f" Precision: {precision:.4f}")
    print(f" Recall   : {recall:.4f}")
```

Evaluation Model : Result 20% Test set

SVM

Accuracy : 0.8136
F1-score : 0.8103
Precision: 0.8171
Recall : 0.8136

KNN

Accuracy : 0.8136
F1-score : 0.8077
Precision: 0.8191
Recall : 0.8136

Decision Tree

Accuracy : 0.7966
F1-score : 0.7900
Precision: 0.7939
Recall : 0.7966

Random Forest

Accuracy : 0.8305
F1-score : 0.8267
Precision: 0.8309
Recall : 0.8305

Model Performance Comparison

Model	CV Accuracy-score	CV F1-score	Test set Accuracy-score	Test set F1-score
SVM	76.06% ($\pm 4.37\%$)	75.93% ($\pm 5.04\%$)	81.36%	81.03%
KNN	71.79% ($\pm 6.80\%$)	72.21% ($\pm 6.81\%$)	81.36%	80.77%
Decision Tree	72.21% ($\pm 5.61\%$)	72.72% ($\pm 5.93\%$)	79.66%	79%
Random Forest	76.92% ($\pm 3.26\%$)	76.71% ($\pm 4.45\%$)	83.05%	82.67%

The Best Model is
Random forest

Cross-validation Results (5-fold) train set - Random Forest
Accuracy : [78.72 74.47 76.6 78.72 76.09] Mean = 76.92% (+/- 3.26%)
F1 : [78.89 73.45 76.68 79.34 75.16] Mean = 76.71% (+/- 4.45%)
Precision: [79.76 74.56 78.54 82.86 77.53] Mean = 78.65% (+/- 5.43%)
Recall : [78.72 74.47 76.6 78.72 76.09] Mean = 76.92% (+/- 3.26%)

Test Set Performance - Random Forest
Accuracy : 0.8305
F1-score : 0.8267
Precision: 0.8309
Recall : 0.8305

confusion matrix for random forest

		Confusion Matrix for Random Forest (Count and %)				
		badminton	basketball	football	snooker or pool	volleyball
True Label	badminton	9 (75.00%)	1 (8.33%)	0 (0.00%)	1 (8.33%)	1 (8.33%)
	basketball	1 (8.33%)	8 (66.67%)	1 (8.33%)	1 (8.33%)	1 (8.33%)
football	0 (0.00%)	1 (9.09%)	9 (81.82%)	0 (0.00%)	1 (9.09%)	
snooker or pool	0 (0.00%)	0 (0.00%)	0 (0.00%)	11 (100.00%)	0 (0.00%)	
volleyball	1 (7.69%)	0 (0.00%)	0 (0.00%)	0 (0.00%)	12 (92.31%)	
Predicted Label	badminton	basketball	football	snooker or pool	volleyball	

รวมค่าถูกต้องทั้งหมด = $9 + 8 + 9 + 11 + 12 = 49$

จำนวนผู้ติด class = 10

Accuracy $\approx 49 / 59 = 83.05\%$

1. สุนูกาเกอร์ (100%)

- โมเดลจำแนกได้สมบูรณ์แบบ
- เป็นกีฬาเดียวที่ไม่ต้องใช้ความคล่องตัวหรือการกระโดด

→ ทำให้แตกต่างจากกีฬาอื่นอย่างชัดเจน

2. บาสเกตบอล (67%) - สับสนมากที่สุด

- สับสนกับ 4 กีฬา (แบดฯ, พุตบอล, วอลเลย์ อย่างละ 1)
- สาเหตุ: บาสมีลักษณะ ผสมผสาน

→ เล่นเป็นกีฬาเหมือนพุตบอล / วอลเลย์

→ เล่นในร่มเหมือนแบดฯ / สุนูกาเกอร์

→ ต้องกระโดดและมีการปะทะ เหมือนพุตบอล
- แนวทางแก้: เพิ่มจำนวนข้อมูลของบาสเกตบอล

เพื่อให้โมเดลเห็นลักษณะเฉพาะได้มากขึ้น

3. แบดมินตัน vs วอลเลย์บอล

- พบการกำหนดสับสนกัน อย่างละ 1 ครั้ง
- เหตุผล: ทั้งคู่เล่นในร่ม ต้องใช้ความคล่องตัว ไม่มีการปะทะ
- จุดต่าง:

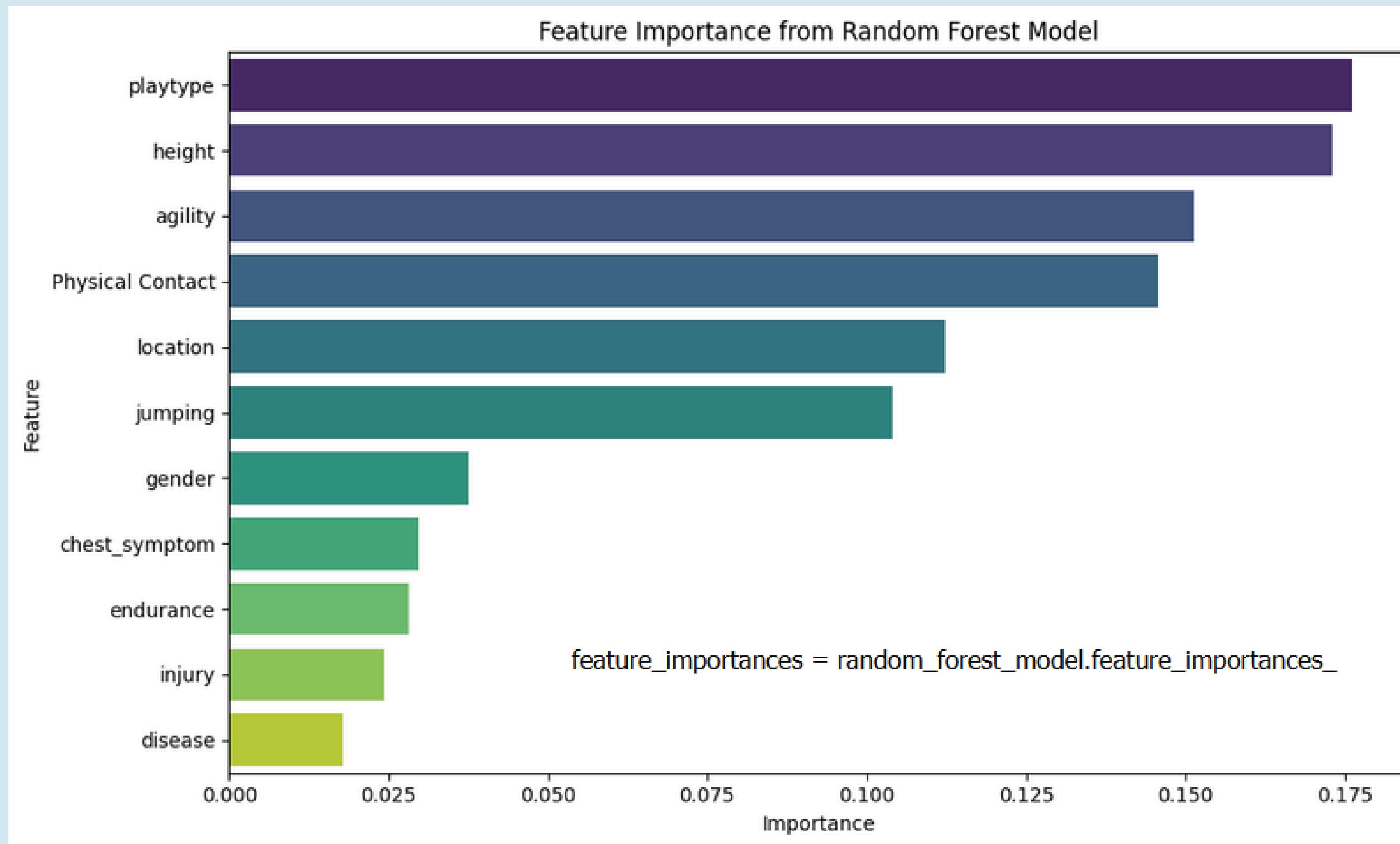
→ รูปแบบการเล่น (เดี่ยว/คู่ vs กีฬา)

→ ส่วนสูงของผู้เล่น (วอลเลย์ต้องการมากกว่า)

Insight from dataset (by visualization or modeling)

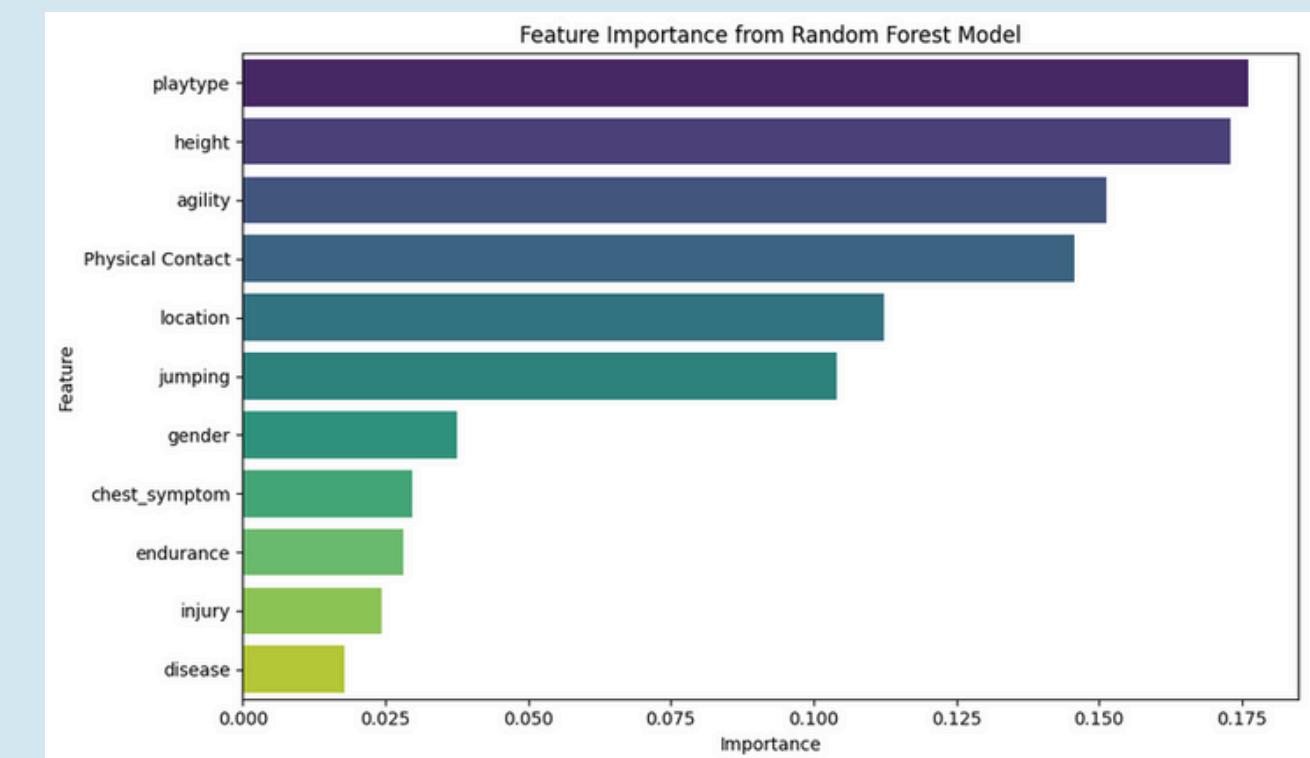


Feature Important From Random Forest

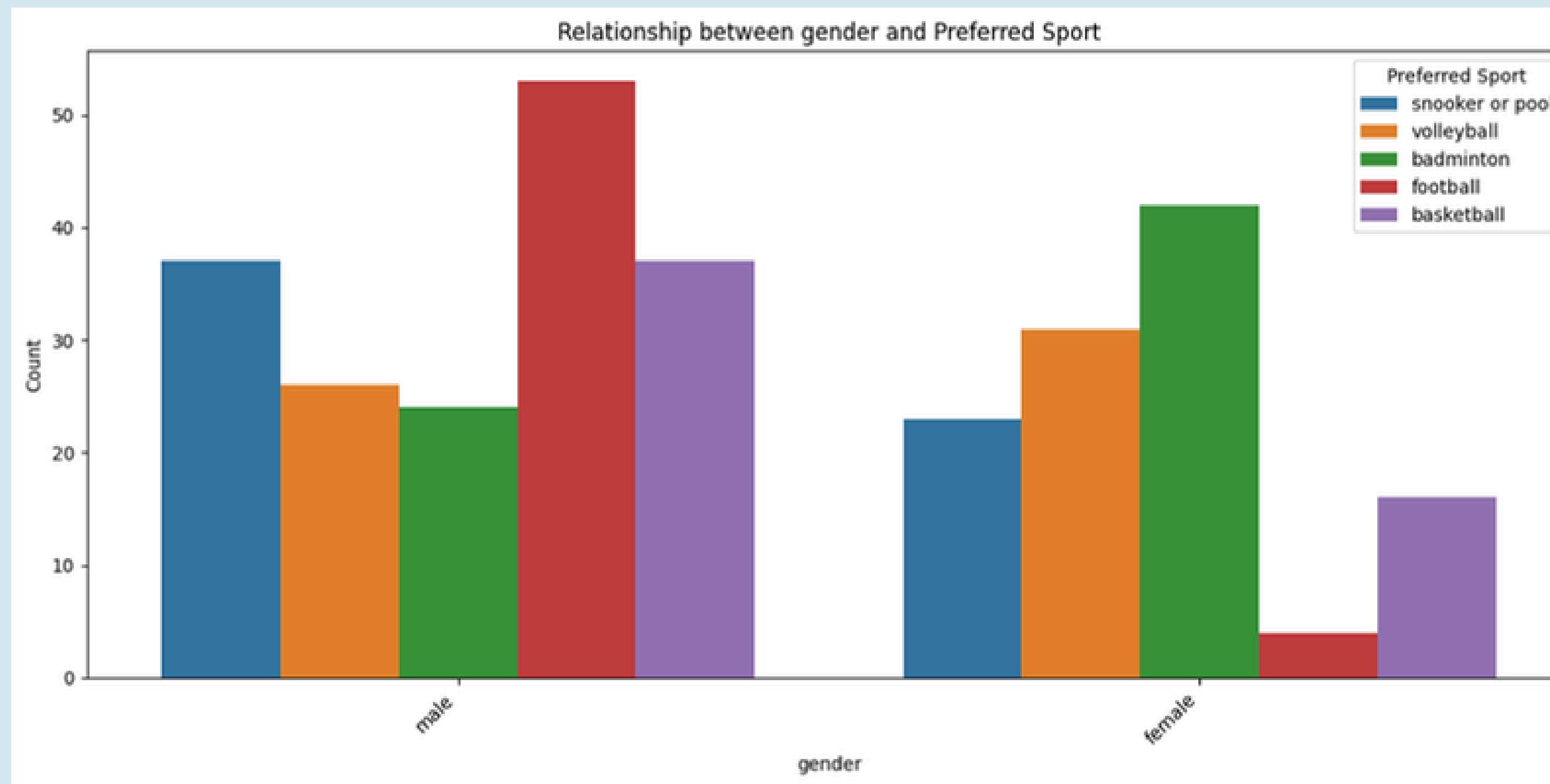


Feature Important From Random Forest(Insight)

1. ปัจจัยสำคัญที่สุด: รูปแบบการเล่น .
playtype (17.6%) - เดี้ยว/คู่/กีฬา → เป็นตัวแบ่งหลักของประเภทกีฬา
2. ลักษณะทางกายภาพ (42.8%) .
height (17.3%) - สูง: บาส/วอลเลย์
agility (15.1%) - ฟุตบอล, แบดมินตัน .
jumping (10.4%) - บาส, วอลเลย์
physical attributes เป็นตัวกำหนดความเหมาะสม
3. ลักษณะการเล่น (25.8%) .
Physical Contact (14.6%) - กีฬาปะทะ vs ไม่ปะทะ .
location (11.2%) - ในร่ม vs กลางแจ้ง
สภาพแวดล้อมและลักษณะกีฬา

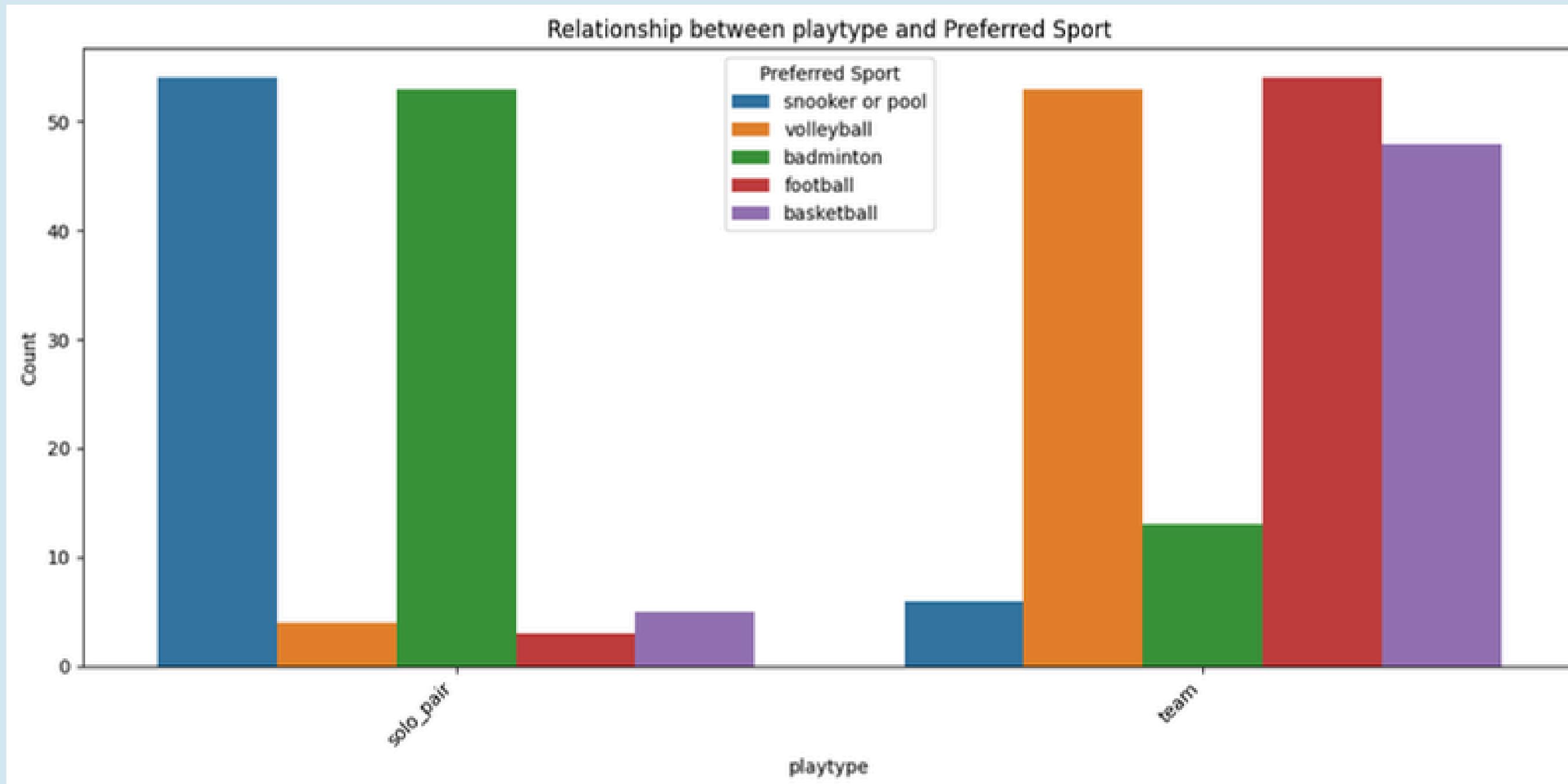


ความสัมพันธ์ระหว่างเพศ กับ กีฬากีฬา



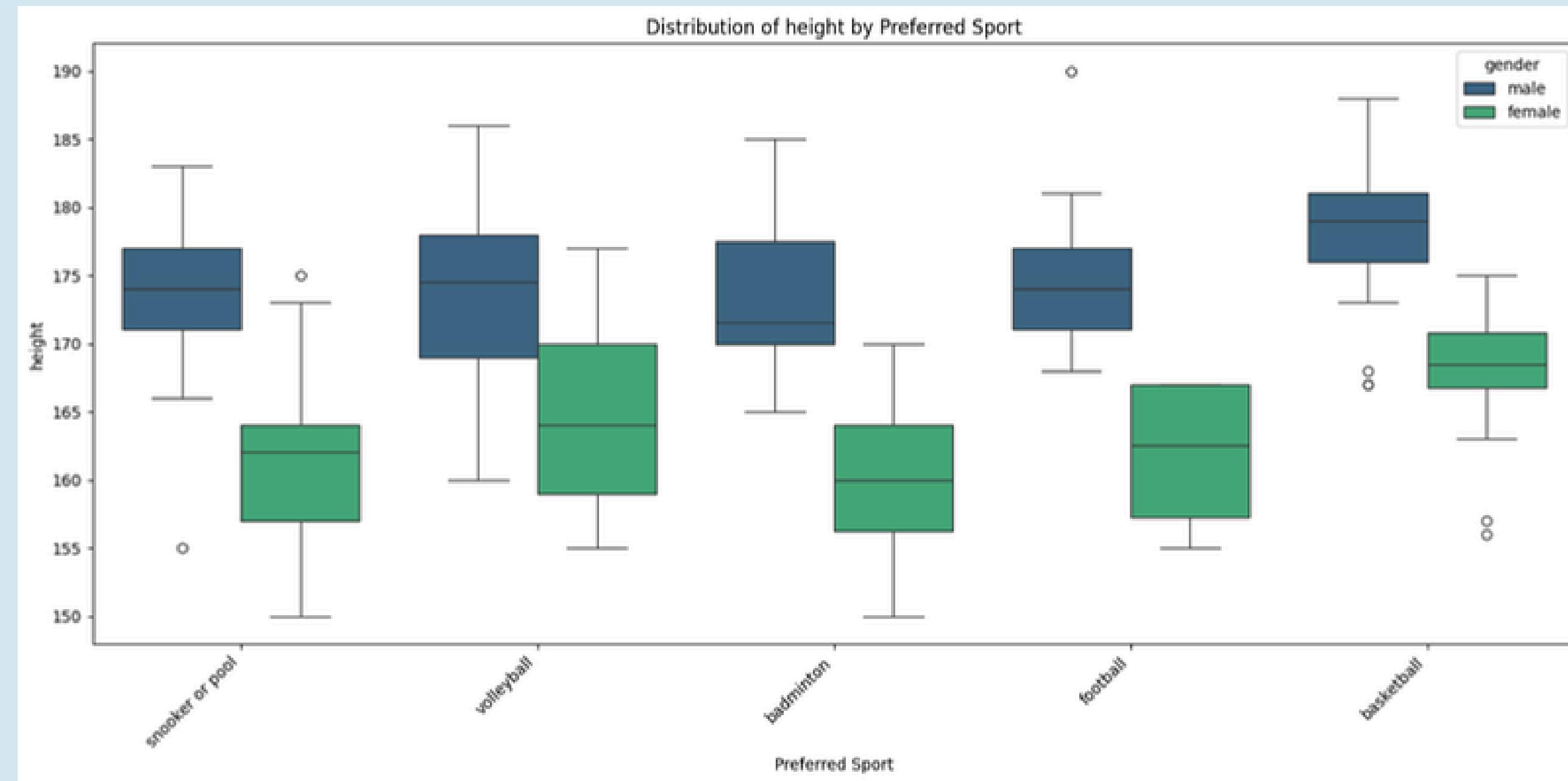
ผู้ชาย สนใจกีฬา พุตบลล์ บาสเกตบอล สนุกเกอร์ มากกว่าผู้หญิง
โดยเฉพาะพุตบลล์ กีฬาที่ผู้หญิงสนใจค่อนข้างน้อย
ผู้หญิง สนใจกีฬา วอลเลย์บลล์ และ แบดมินตัน มากกว่าผู้ชาย

ความสัมพันธ์ระหว่าง ประเภทกีฬา กับ กีฬากีฬา



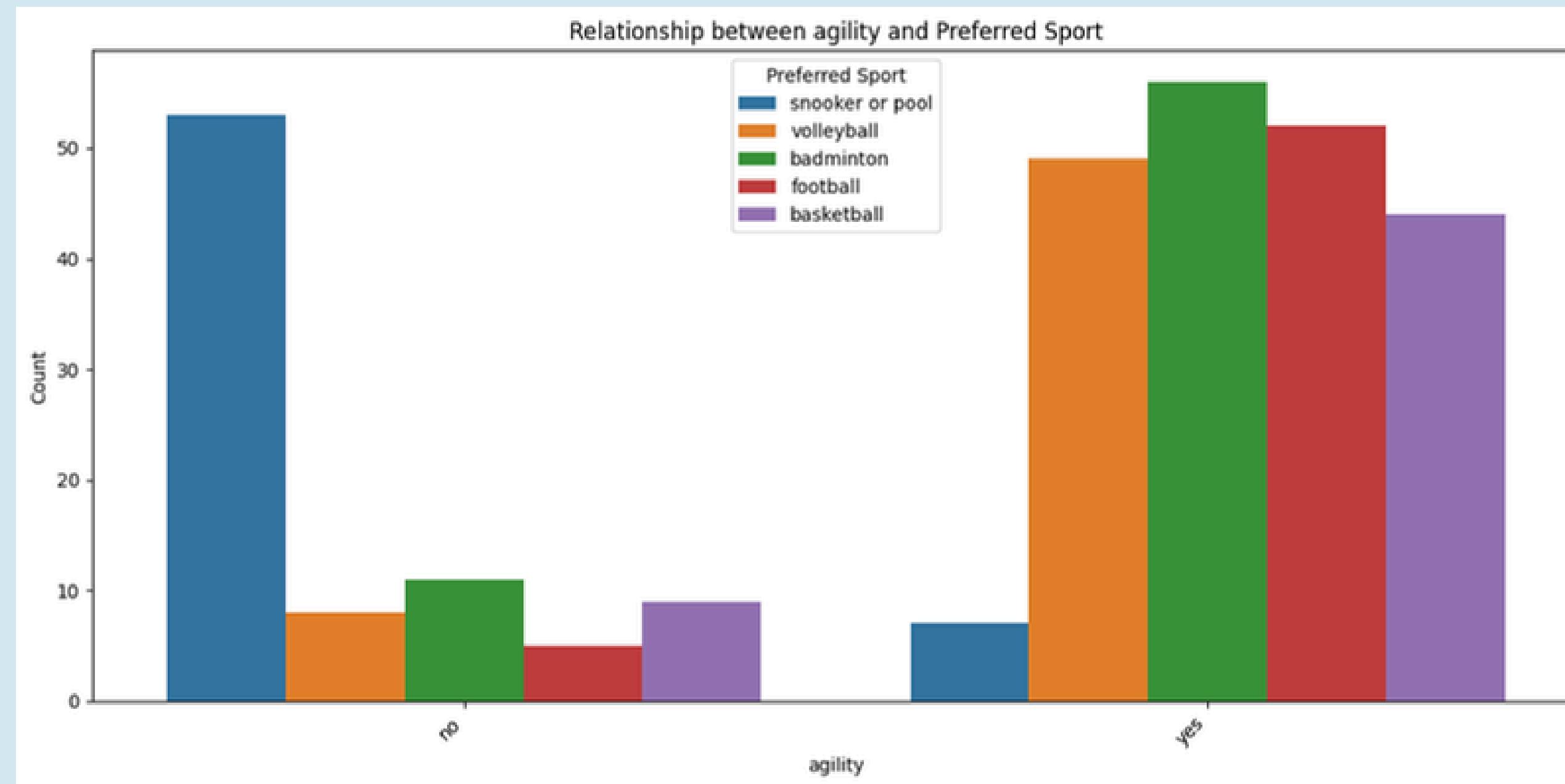
ผู้ที่ชอบประเภท เดี่ยว และ คู่ ส่วนใหญ่เลือก แบดมินตัน และ สนุกเกอร์/พูล
ขณะที่กลุ่มที่ชอบประเภทกีฬา เลือกกีฬา ฟุตบอล, บาสเกตบอล, วอลเลย์บอล เกือบกึ่งหนึ่ง

ความสัมพันธ์ระหว่าง ส่วนสูง กับ กีฬาที่ชอบโดยแยกตามเพศ



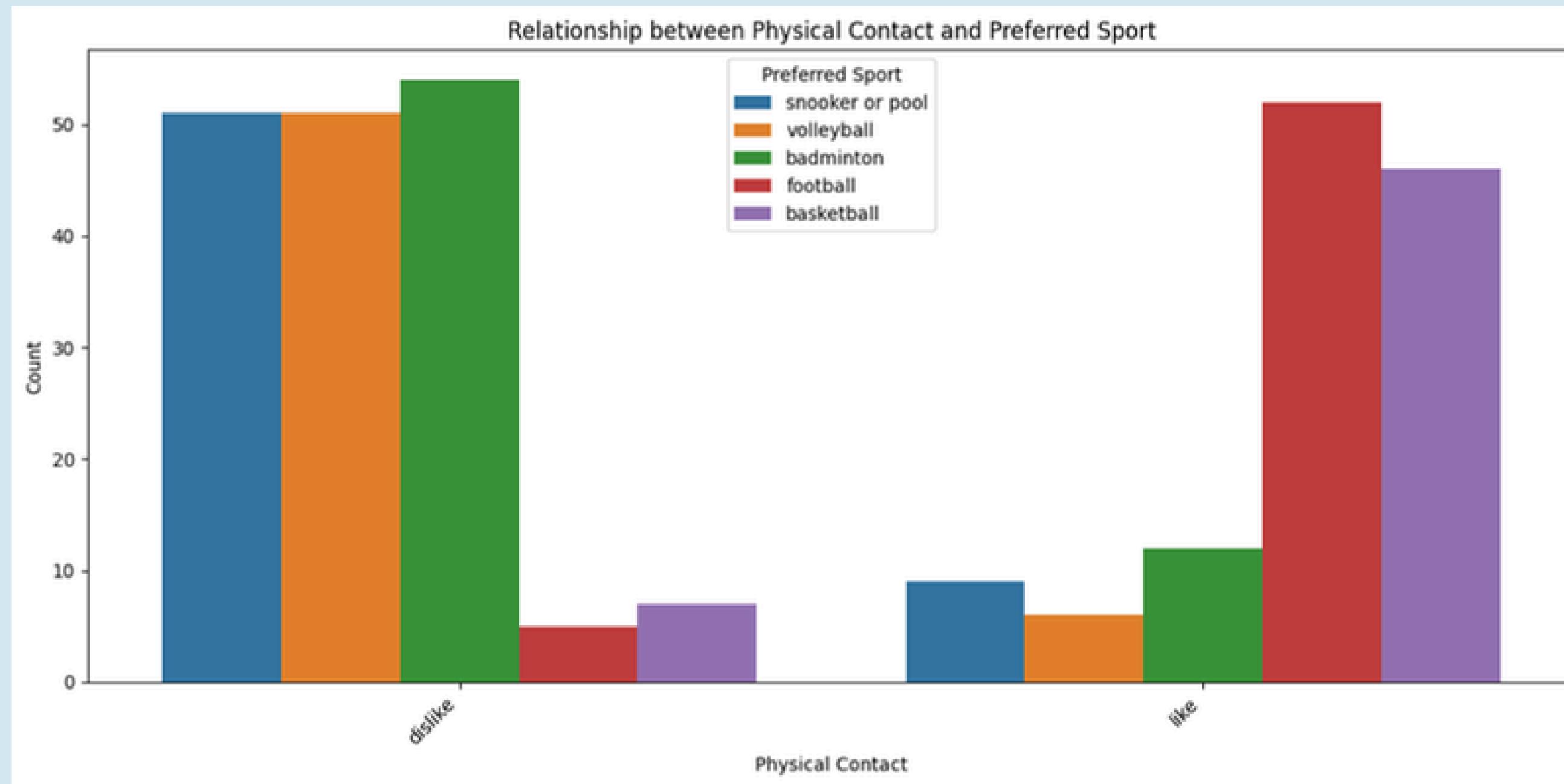
ก็งในเพศชายและหญิง ส่วนสูงกลุ่มผู้ที่ชอบ บาส/วอลเลย์ มีค่า มัธยฐานและ ควรไกล์บัน สูงกว่า แบดมินตัน /สุนูกเกอร์ อย่างมีนัยสำคัญ ขณะที่ พุตราล อยู่กลาง ๆ แสดงว่า ส่วนสูงเป็นสัญญาณสำคัญในการประกอบกีฬา

ความสัมพันธ์ระหว่าง ผู้ที่ชอบกิจกรรมที่ใช้ความคล่องตัว กับ กีฬาที่ชอบ



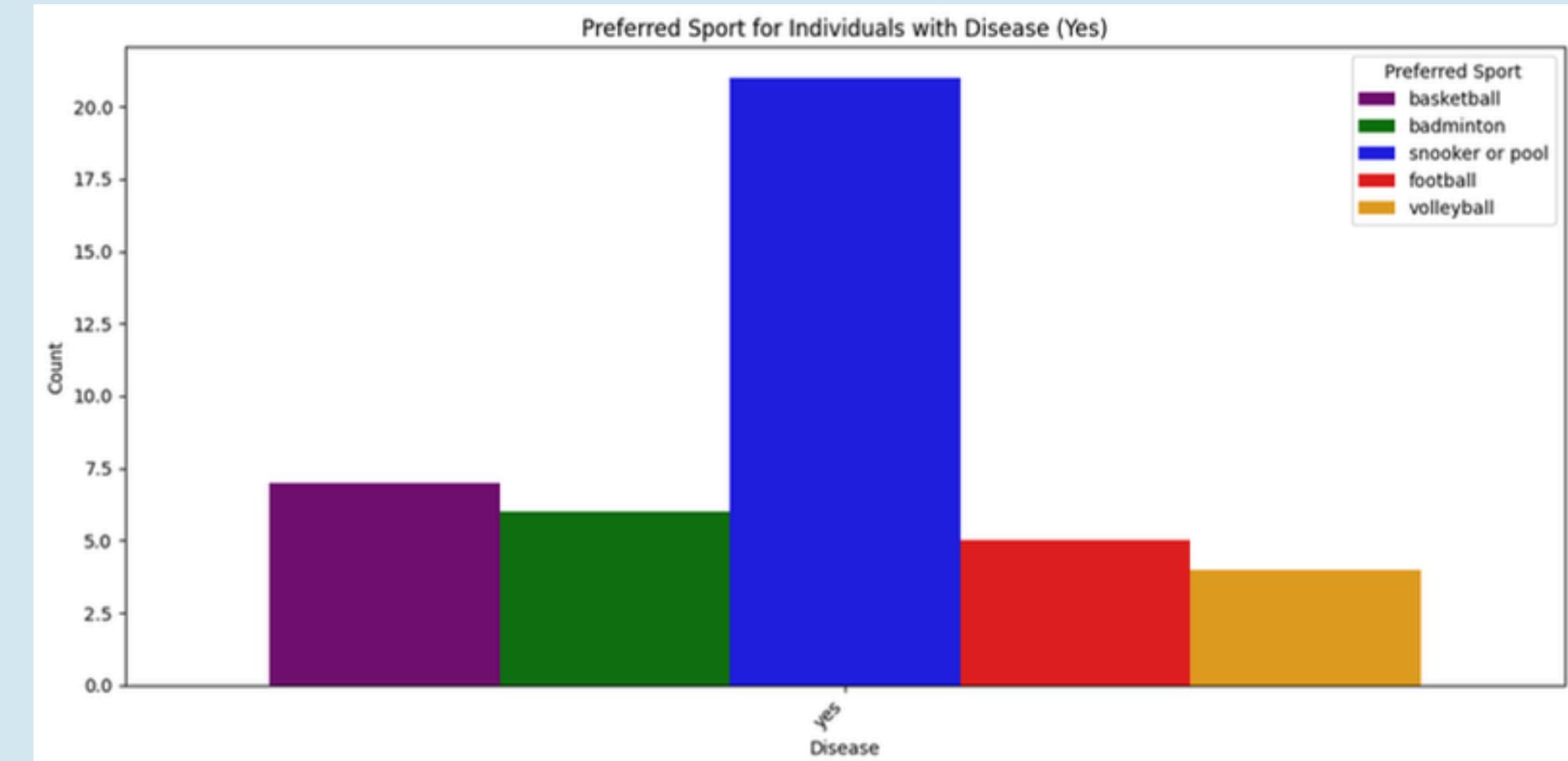
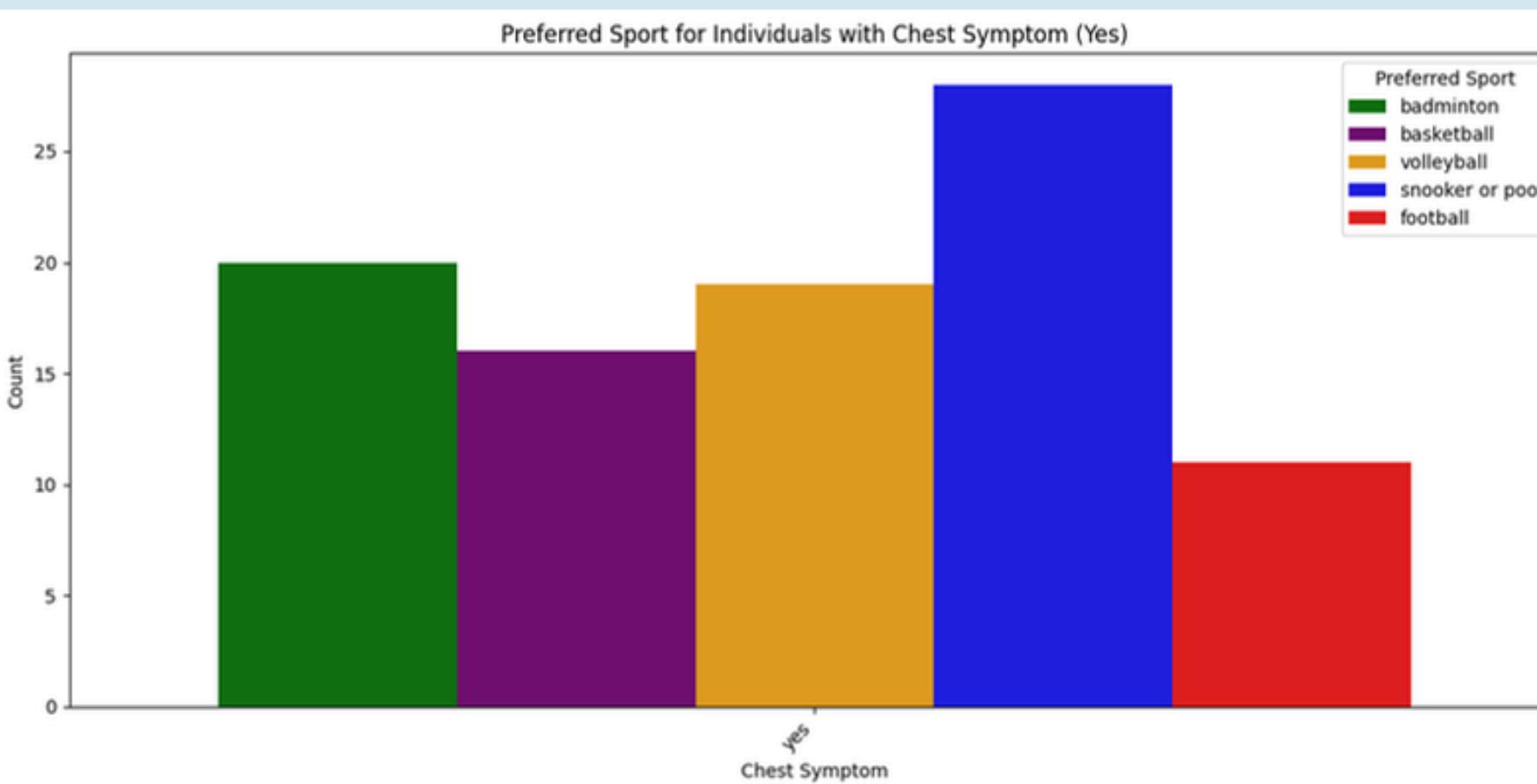
ผู้ที่ชอบกิจกรรมที่ใช้ความคล่องตัว มักเลือก วอลเลย์บอล,
แบดมินตัน, ฟุตบอล, บาสเกตบอล ขณะที่ สนุกเกอร์/พูล เป็น
ตัวเลือกของกลุ่มที่ไม่ชอบใช้ความคล่องตัว

ความสัมพันธ์ระหว่างการปะทะกับกีฬาที่ชอบ



ผู้ที่ชอบเล่นสนุกเกอร์ วอลเลย์บอล และแบดมินตันส่วนใหญ่ไม่ชอบการปะทะร่างกาย ผู้ที่ชอบการปะทะร่างกายมากจะสนใจฟุตบอลและบาสเกตบอลเป็นหลัก

ความสัมพันธ์ระหว่างประวัติโรคทางเดินหายใจ และ อาการ แน่นหน้าอกระหว่างกำกิจกรรม กับ กีฬาที่ชอบ



ผู้ที่มีข้อจำกัดทางสุขภาพ (ไม่ว่าผู้ที่เคยมีประวัติโรคปอด และหัวใจ, อาการแน่นหน้าอกระหว่างกำกิจกรรม) จะหลีกเลี่ยง กีฬาที่ต้องใช้กำลังและความทนทานสูง (เช่น พุตบอล, วอลเลย์บอล, แบดมินตัน) อย่างชัดเจน และเลือกซ้อม “สนุกเกอร์/พูล” มากที่สุด ซึ่งเป็นกิจกรรมที่ใช้กำลังกายน้อยที่สุดในบรรดา กีฬาที่สำรวจ

Problem and suggestion

1) Outlier ของน้ำหนัก (Weight) มาก และไม่ให้ insight ต่อ กีฬา

Problem: เพิ่ม noise/เสียง overfit และไม่ช่วยจำแนก กีฬา

Action: ตัดฟีเจอร์ weight ออกจาก การ เทคนิค

Suggestion: ถ้าเก็บข้อมูลได้เยอะมากขึ้น และ weight หา insight ในอนาคต ได้ลองเลือก การ scaling ข้อมูลด้วยวิธีกึ่งต่อ outlier เช่น robust scaling (median/IQR)

2). random forest ให้ความสำคัญกับ feature ที่เกี่ยวกับสภาพร่างกาย น้อยมาก

Problem:

ผู้ตอบที่มีประวัติโรคทางเดินหายใจ หรือ อาการบาดเจ็บในปัจจุบัน มีจำนวนค่อนข้างน้อย
ก้าวให้โมเดล Random Forest ประเมินว่า ฟีเจอร์ Disease และ Injury มีความสำคัญต่ำ
 เพราะข้อมูลในกลุ่มนี้ไม่เพียงพอให้โมเดลเรียนรู้ความแตกต่างอย่างชัดเจน

Suggestion: เพิ่มการเก็บข้อมูลโดยเฉพาะกลุ่มที่มี

- ประวัติเป็นโรคเกี่ยวกับระบบทางเดินหายใจ (เช่น หอบหืด, ปอดอักเสบ)
- อาการบาดเจ็บ เช่น ข้อเท้า หรือ เข่าในปัจจุบัน
- เพื่อให้โมเดลเข้าใจผลของ “สภาพร่างกาย” ต่อการเลือกประเภทกีฬาได้ดียิ่งขึ้น

	Feature	Importance
10	playtype	0.176106
0	height	0.172985
9	agility	0.151397
3	Physical Contact	0.145622
2	location	0.112258
7	jumping	0.104075
1	gender	0.037583
4	chest_symptom	0.029661
8	endurance	0.028131
5	injury	0.024395
6	disease	0.017785

MEMBERS

1. 66050158 ເດືອນ ຊັກ ດີ້ງາມ
2. 66050168 ກິພວຮຣະນ ເຂີຍຈານ
3. 66050172 ຮນກຖຕີ ອັກຊຣເຈຣີລູສຸຂ
4. 66050191 ຮຣາເກພ ຈູໂຈຣີລູ
5. 66050208 ນນກການຕໍ ເລີສກຮ້ພຍ໌ຈິນດາ
6. 66050298 ພັກຮດນຍ໌ ຂອງໃນຂາວ
7. 66050406 ສຸກກິຈ ຕຳ້ງທັກຍກີພຍ໌
8. 66050412 ສຸກສຣາ ໄກວ່າກວີ