牛刀小试——足球运动员分析

背景信息

当前，足球运动是最受欢迎的运动之一（也可以说没有之一）。在此万受瞩目的运动下，我们打算针对足球运动员个人的信息，技能水平等各项指标进行相关的分析与统计。例如，我们可能会关注如下的内容：

- 足球运动员是否受出生日期的影响？
- 左撇子适合踢足球吗？
- 足球运动员的号码是否与位置相关？
- 足球运动员的年龄与能力具有怎样的关联？
- 哪些技能会对足球运动员的综合能力造成较大的影响？

任务说明

目前，我们收集到了2019年现役运动员的数据集。我们希望通过该数据集，针对众多的足球运动员进行分析与统计，从而能够发现一些关于足球运动员的特征，解开之前的谜题。

数据集描述

数据集包含的是2019年现役的足球运动员。

列名含义

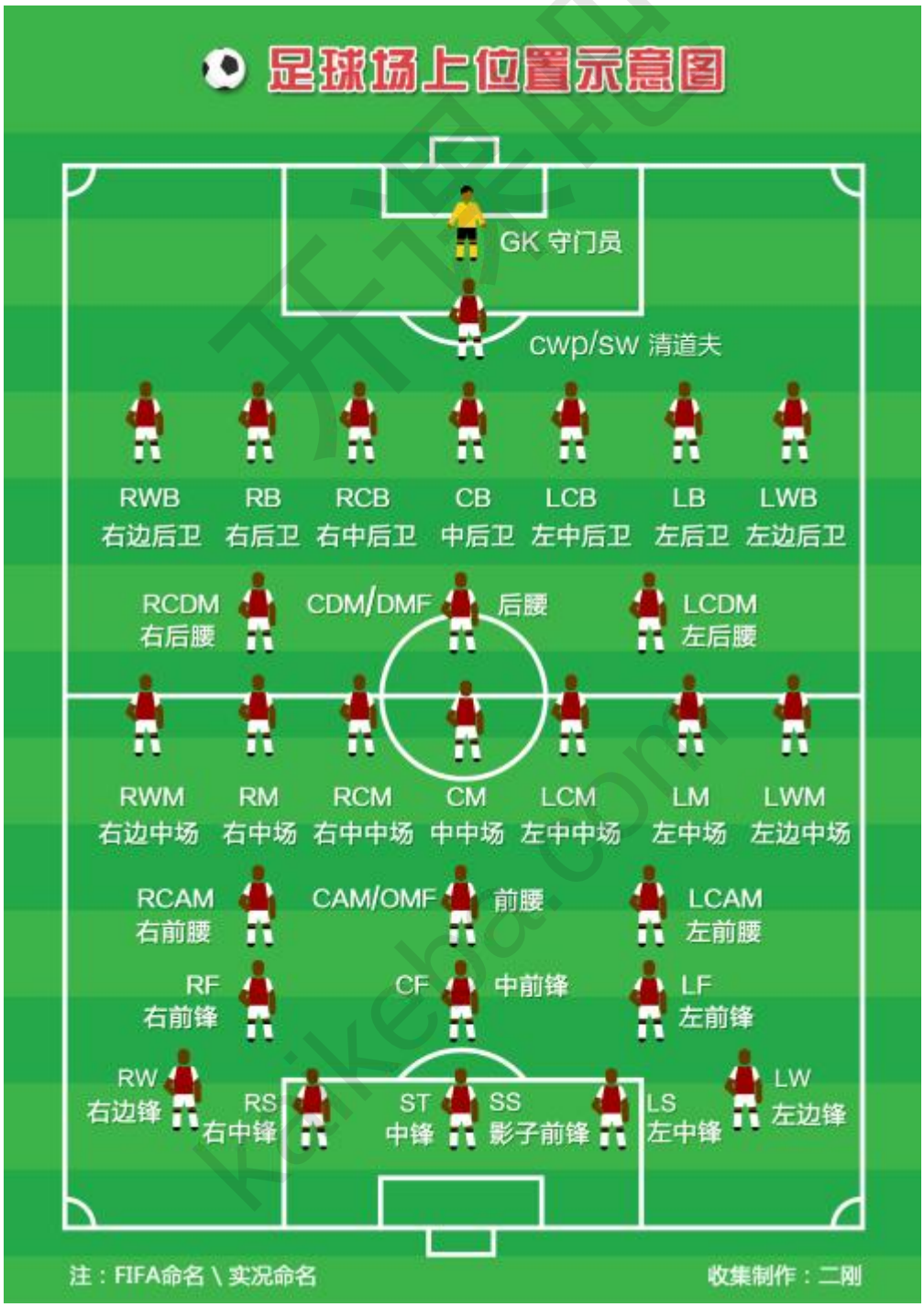
列名	含义

列名	含义
Name	球员姓名
Age	年龄
Nationality	国籍
Overall	综合能力评分
Potential	潜能评分
Club	所属俱乐部
Value	球员身价
Wage	周薪
Preferred Foot	惯用脚
Position	最佳位置
Jersey Number	运动衫号码
Joined	加入俱乐部时间
Height	身高
Weight	体重
Crossing	传中
Finishing	射术
HeadingAccuracy	头球精度
ShortPassing	短传
Volleys	凌空
Dribbling	盘带
Curve	弧线
FKAccuracy	任意球精度
LongPassing	长传
BallControl	控球
Acceleration	加速
SprintSpeed	速度
Agility	敏捷
Reactions	反应
Balance	平衡
ShotPower	射门力量
Jumping	弹跳
Stamina	体能
Strength	强壮
LongShots	远射
Aggression	侵略性
Interceptions	拦截意识
Positioning	跑位
Vision	视野
Penalties	点球
Composure	沉着
Marking	盯人
StandingTackle	抢断
SlidingTackle	铲球
GKDiving	鱼跃
GKHandling	手形
GKKicking	开球
GKPositioning	站位

列名	含义
GKReflexes	反应
Release Clause	违约金

场上位置

场上位置如下图所示（图片来自互联网）。



数据分析

概念

数据分析是指用适当的方法与工具，对收集来的大量数据进行分析，提取其中有意义的信息，从而形成有价值的结论的过程。

基本流程

在进行数据分析之前，我们需要清楚数据分析的基本流程。

- 明确需求与目的
- 数据收集
    - 内部数据
    - 购买数据
    - 爬取数据
    - 调查问卷
- 数据预处理
    - 数据清洗
        - 缺失值
        - 异常值
        - 重复值
    - 数据转换
- 数据分析
    - 数据建模
    - 数据可视化
- 编写报告

理解流程

我们可以进行一个类比，比如，我们现在要做出一道菜肴，那我们需要进行怎样的流程呢？

- 确定做菜
- 买菜
- 洗菜
- 切菜
- 炒菜
- 盛菜
- 写日记

接下来，我们就可以将做菜的流程步骤去对应理解数据分析的步骤。

做菜步骤	**数据分析步骤**
确定做菜	明确需求与目的
买菜	数据收集
洗菜	数据清洗
切菜	数据转换
炒菜	数据分析
盛菜	数据可视化
写日记	编写报告

程序实现

导入相关的库

导入需要的库，同时，进行一些初始化的设置。

```
import numpy as np
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
import warnings

# mpl.rcParams["font.family"] = "SimHei"
# mpl.rcParams["axes.unicode_minus"] = False

import seaborn as sns
sns.set(style="darkgrid", font="SimHei", font_scale=1.5, rc={"axes.unicode_minus": False})
warnings.filterwarnings("ignore")
```

加载相关的数据集

- 加载相关的数据集。
- 可以使用head / tail / sample查看数据的大致情况。

```
# 读取参数指定的文件，返回一个DataFrame类型的对象。
data = pd.read_csv("data.csv")
print(data.shape)
data.head()
# data.tail()
# data.sample()
```

不过，数据集中的列，并非都是我们分析所需要的，我们可以有选择性的进行加载，只加载我们需要的信息列（特征列）。

```
columns = ["Name", "Age", "Nationality", "Overall", "Potential", "Club", "Value", "Wage",
"Preferred Foot",
           "Position", "Jersey Number", "Joined", "Height", "Weight", "Crossing", "Finishing",
           "HeadingAccuracy", "ShortPassing", "Volleys", "Dribbling", "Curve", "FKAccuracy",
"LongPassing",
           "BallControl", "Acceleration", "SprintSpeed", "Agility", "Reactions", "Balance",
"ShotPower",
           "Jumping", "Stamina", "Strength", "LongShots", "Aggression", "Interceptions",
"Positioning", "Vision",
           "Penalties", "Composure", "Marking", "StandingTackle", "SlidingTackle", "GKDiving",
"GKHandling",
           "GKKicking", "GKPositioning", "GKReflexes", "Release Clause"]
# 参数指定所要读取的列。
data = pd.read_csv("data.csv", usecols=columns)
data.head()

# 设置显示的最大列数。
pd.set_option("max_columns", 100)
data.head()
```

数据清洗

缺失值处理

- 通过info查看数据信息。

- 可以通过isnull与sum结合，查看缺失值情况。

```
# info方法可以显示每列名称，非空值数量，每列的数据类型，内存占用等信息。
data.info()
# data.isnull().sum(axis=0)

# 删除所有含有空值的行。就地修改。
data.dropna(axis=0, inplace=True)
data.isnull().sum()
```

异常值处理

- 通过describe查看数值信息。
- 可配合箱线图辅助。
- 异常值可以删除，视为缺失值，或者不处理。

```
data.describe()
# sns.boxplot(data=data[["Age", "Overall"]])
```

重复值处理

- 使用duplicate检查重复值。可配合keep参数进行调整。
- 使用drop_duplicate删除重复值。

```
data.duplicated().sum()
# data.drop_duplicates(inplace=True)
```

数据分析

足球运动员的身高体重分布。

数据转换

我们要统计身高与体重的分布情况，不过，身高与体重目前并不是数值类型，我们需要进行转换后，才能进行统计计算。这里，我们将身高与体重转换成熟悉的单位。
1英尺 = 30.48厘米
1英寸 = 2.54厘米
1磅 = 0.45千克

```
# 定义转换函数
def tran_height(height):
    v = height.split("'")
    return int(v[0]) * 30.48 + int(v[1]) * 2.54

def tran_weight(weight):
    v = int(weight.replace("lbs", ""))
    return v * 0.45

data["Height"] = data["Height"].apply(tran_height)
data["Weight"] = data["Weight"].apply(tran_weight)
```

绘制核密度图

数据转换后，我们可以来绘制下身高与体重的分布。

```
fig, ax = plt.subplots(1, 2)
fig.set_size_inches((18, 5))
sns.distplot(data[["Height"]], bins=50, ax=ax[0], color="g")
sns.distplot(data["Weight"], bins=50, ax=ax[1])
```

左撇子适合踢足球吗?

数量上对比

我们首先从球员数量上进行一下统计。

```
number = data["Preferred Foot"].value_counts()
print(number)
sns.countplot(x="Preferred Foot", data=data)
```

能力上对比

然后，我们再从球员综合能力上进行衡量。

```
print(data.groupby("Preferred Foot")["Overall"].mean())
sns.barplot(x="Preferred Foot", y="Overall", data=data)
```

位置上对比

由于在综合能力上体现不明显，我们现在通过每个位置，进行更细致的分析。为了分析的客观性，我们只统计左脚与右脚都超过50人（含50人）的位置。
首先，我们来计算哪些位置左右脚球员都达到了50人。

```
t = data.groupby(["Preferred Foot", "Position"]).size()
t = t.unstack()
t[t < 50] = np.NaN
t.dropna(axis=1, inplace=True)
display(t)
```

然后，我们根据之前计算的那些位置，对数据集进行过滤。

```
t2 = data[data["Position"].isin(t.columns)]
plt.figure(figsize=(18, 10))
sns.barplot(x="Position", y="Overall", hue="Preferred Foot", hue_order=["Left", "Right"],
data=t2)
```

从结果可以清晰得知，左脚选手更适合RW(右边锋）的位置。

哪个的俱乐部 / 国家拥有综合能力更好的球员（top10）。

由于每个俱乐部/国家队人数不一，为了统计的客观性，只考虑人数达到一定规模的俱乐部/国家。

俱乐部

```
g = data.groupby("Club")
r = g["Overall"].agg(["mean", "count"])
r = r[r["count"] >= 20]
r = r.sort_values("mean", ascending=False).head(10)
display(r)
r.plot(kind="bar")
```

国家队

```
g = data.groupby("Nationality")
r = g["Overall"].agg(["mean", "count"])
r = r[r["count"] >= 50]
r = r.sort_values("mean", ascending=False).head(10)
display(r)
r.plot(kind="bar")
```

哪个俱乐部拥有效力更久的球员（5年及以上）？

```
t = pd.to_datetime(data["Joined"])
t = t.astype(np.str)

join_year = t.apply(lambda item: int(item.split("-")[0]))
over_five_year = (2018 - join_year) >= 5
t2 = data[over_five_year]
t2 = t2["Club"].value_counts()
# display(t2)
t2.iloc[:15].plot(kind="bar")
```

足球运动员是否是出生日期相关?

我们现有的数据集中，不含有具体的出生日期，因此，我们使用另外一个数据集，该数据集包含2018年世界杯所有球员。

```
data2 = pd.read_csv("wc2018-players.csv")
data2.head()

t = data2["Birth Date"].str.split(".", expand=True)
t[0].value_counts().plot(kind="bar")
# t[1].value_counts().plot(kind="bar")
# t[2].value_counts().plot(kind="bar")
# t[2].value_counts().sort_index().plot(kind="bar")
```

足球运动员号码是否与位置相关?

```
g = data.groupby(["Jersey Number", "Position"])
t = g.size()
display(t)
t = t[t >= 100]
t.plot(kind="bar")
```

身价与薪水，违约金是否相关?

因为身价与违约金的单位既有M，也有K，我们统一K单位，同时，将类型转换为数值类型，便于统计。

```
def to_numeric(item):
    item = item.replace("€", "")
    value = float(item[:-1])
    if item[-1] == "M":
        value *= 1000
    return value

data["Value"] = data["Value"].apply(to_numeric)
data["Wage"] = data["Wage"].apply(to_numeric)
data["Release Clause"] = data["Release Clause"].apply(to_numeric)
data.head()
```

```
# sns.scatterplot(x="Value", y="Wage", data=data)
# sns.scatterplot(x="Value", y="Release Clause", data=data)
# sns.scatterplot(x="Value", y="Height", data=data)
```

哪些指标对综合评分的影响较大?

```
# data.corr()
plt.figure(figsize=(25, 25))
sns.heatmap(data.corr(), annot=True, fmt=".2f", cmap=plt.cm.Greens)
plt.savefig("corr.png", dpi=100, bbox_inches="tight")
```

分析某项未标记的技能

假设因为某种原因，GKDiving列的标题没有成功获取，现在分析该技能可能表示的含义。

```
g = data.groupby("Position")
g["GKDiving"].mean().sort_values(ascending=False)
plt.figure(figsize=(15, 5))
sns.barplot(x="Position", y="GKDiving", data=data)
```

年龄与评分具有怎样的关系?

```
sns.scatterplot(x="Age", y="Overall", data=data)

data["Age"].corr(data["Overall"])

# 对一个数组进行切分，可以将连续值变成离散值。
# bins 指定区间数量（桶数）。bins如果为int类型，则进行等分。
# 此处的区间边界与为前开后闭。
# pd.cut(t["Age"], bins=4)
# 如果需要进行区间的不等分，则可以将bins参数指定为数组类型。
# 数组来指定区间的边界。
min_, max_ = data["Age"].min() - 0.5, data["Age"].max()
# pd.cut(t["Age"], bins=[min_, 20, 30, 40, max_])
# pd.cut 默认显示的内容为区间的范围，如果我们希望自定义内容(每个区间显示的内容)，可以通过labels参数
# 进行指定。
t = pd.cut(data["Age"], bins=[min_, 20, 30, 40, max_], labels=["弱冠之年", "而立之年","不惑之年",
"知天命"])
t = pd.concat((t, data["Overall"]), axis=1)
g = t.groupby("Age")
display(g["Overall"].mean())
sns.lineplot(y="Overall", marker="*", ms=30, x="Age", data=t)
```

总结

1. 左撇子相对于右撇子来说，并无明显劣势，其更适合右边锋的位置。
2. 知名俱乐部平均能力更好的球员，但并非球员平均能力越好，球队的成绩就越好。
3. 一些知名足球国家，在球员的平均能力上可能并没有非常靠前，只是因为足球运动员较多，进而个别球员较知名而已。
4. 足球运动员与出生日期是有关的，在年初出生的运动员要明显多于在年末出生的运动员。
5. 足球运动员的号码与位置是相关的，例如，1号通常都是守门员，9号通常是中锋等。
6. 足球运动员的身价与其薪水是紧密关联的，尤其是违约金，与身价的关联更大。
7. Reactions（反应）与Composure（沉着）两项技能对总分的影响最大。
8. 随着年龄的增长，球员得到更多的锻炼与经验，总体能力提升，但三十几岁之后，由于体力限制，总体能力下降。