1. 字段的选取
2. pd.set\_option(“max\_columns”, 50) 设置最大显示字段数量
3. 缺失值处理
4. 各字段有多少条缺失记录 data.isnull().sum(axis=0)
5. 含有缺失字段的记录条数 data.isnull().sum(axis=1).value\_counts()
6. 确定缺失值的处理方法

若删除含缺失值的记录，可通过inplace参数确定是否替换原始对象

inplace=True，修改原始对象

inplace=False：对数据进行修改，创建并返回新的对象承载其修改结果

1. 异常值处理
2. data.describe()
3. 绘制箱线图

import seaborn as sns

sns.set(style=”darkgrid”, font=”SimHei”, font\_scale=1.5, rc={“axes.unicode\_minus”:False})

sns.boxplot(data=[[“aa”,”bb”]])

sns.boxplot(x=””, y=””, data=)

显示sns绘图结果：import matplotlib.pyplot as plt

plt.show()

<http://liyangbit.com/pythonvisualization/Data-Visualization-with-Seaborn/>

③ 可删除、视为缺失值、或不处理

1. 重复值处理

标记重复值：data.duplicated().sum()

若想直接删除：data.drop\_duplicates(inplace=True)

标记或删除重复值，可添加keep参数

keep=’frist’：除了第一次出现外，其余相同的被标记为重复(默认)

keep=’last’：除了最后一次出现外，其余相同的被标记为重复

keep=False：所有相同的都被标记为重复

1. 绘制图形

绘制核密度图

fig, ax = plt.subplots(1,2)

fig.set\_size\_inches((18,5))

sns.distplot(data[[“ ”]], bins=50, ax=ax[0], color=”g”)

plt.savefig(‘’,dpi=)

<https://zhuanlan.zhihu.com/p/24464836>

1. 绘制计数图、条形图，条形图y轴表示的是均值
2. data.groupby([“”, “”]).size() 返回分组字段的计数DataFrame

unstack() 作用于内层索引，行转为列

stack() 列转为行

1. 同列运用多个函数，可使用agg，同一函数运用到多列，可使用apply

file.groupby(“”).agg([“”,””])

<https://blog.csdn.net/qq_16234613/article/details/78245325>

1. 切片
   1. loc 通过行标签索引行数据，左闭右闭
   2. iloc 通过行号索引行数据，左闭右开
2. pd.read\_csv报错,
   1. 在调用read\_csv()方法时指定engine=”python”
   2. 使用open函数打开文件，再取访问里面的数据 data=pd.read\_csv(open(“”))

<https://blog.csdn.net/qq_35318838/article/details/80564938>

1. str 属性只对series

split的expand=True参数将字符串拆分成多列，返回一个数据框

str.split()默认从左向右拆分，str.rsplit()从右往左拆分，n=1参数指定拆分次数

<https://www.jianshu.com/p/31daa943cd2b>

1. 绘制热图

sns.heatmap(data, annot=True, fmt=”.2f”, camp=plt.cm.Greens)

annot——是否在小格子中显示数据，fmt——数据显示类型，camp图形颜色(RdYlGn，数值越小越红，若要越大越红，则在颜色后加上”\_r”，即RdYlGn\_r)

1. 保存图片

plt.savefig(“name”, dpi=100, bbox\_inches=”tight”)

bbox\_inches=”tight”——表示去掉多余的白边

1. 绘制折线图

sns.lineplot(x=””, y=””, ci=”sd”, data=)

ci默认以置信区间绘图，=“sd”表示以标准差绘图

1. 散点图

若x轴、y轴均为连续型变量，可用sns.scatterplot，hue参数可增加一个维度，palette参数设置颜色

若x为离散型变量(分类变量)、y为连续型变量，可用sns.swarmplot

1. t检验

from scipy import stats

stats.ttest\_lsamp(data[“AQI”], 71)

mean - 1.96 \* (std / np.sqrt(len(data))), mean + 1.96 \* (std / np.sqrt(len(data)))

<https://blog.csdn.net/Peter_Luoz/article/details/80872392>

1. 线性回归

from sklearn.linear\_model import LinearRegression

from sklearn.model\_selection import train\_test\_split

X = data.drop(['city', 'AQI'], axis=1)

y = data['AQI']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.25, random\_state=0)

lr = LinearRegression()

lr.fit(X\_train, y\_train)

y\_hat = lr.predict(X\_test)

print(lr.score(X\_train, y\_train))

print(lr.score(X\_test, y\_test))

plt.figure(figsize=(15, 5))

plt.plot(y\_test.values, "-r", label="真实值")

plt.plot(y\_hat, "-g", label="预测值")

plt.legend()

plt.title("线性回归预测结果")

plt.show()

<https://blog.csdn.net/sinat_36219858/article/details/79800460>

1. np.argmax()

axis=1：按行查找最大元素

axis=0：按列查找最大元素

<https://blog.csdn.net/u013713117/article/details/53965572>

1. 堆积条形图

index = np.arange(len(X\_test))

pro\_0 = probability[:, 0]

pro\_1 = probability[:, 1]

tick\_label = np.where(y\_test == y\_hat, "O", "X")

plt.figure(figsize=(15, 5))

plt.bar(index, height=pro\_0, color="g", label="类别0概率值")

plt.bar(index, height=pro\_1, color="r", bottom=pro\_0, label="类别1概率值", tick\_label=tick\_label)

plt.legend(loc="best", bbox\_to\_anchor=(1, 1))

plt.xlabel("样本序号")

plt.ylabel("各个类别的概率")

plt.title("逻辑回归分类概率")

loc表示图例放置位置，bbox\_to\_anchor表示放在图形边界外

<https://blog.csdn.net/helunqu2017/article/details/78641290>

可考虑把数据转成DataFrame，然后画堆积条形图df.plot(kind=”bar”, stacked=True)

1. plt.subplot2grid((2,3),(0,0))

<https://morvanzhou.github.io/tutorials/data-manipulation/plt/4-2-subplot2/>

1. 解决中文乱码问题

<https://www.zhihu.com/question/25404709>

