

学习周报

Yong YANG

Beijing University of Posts and Telecommunications
<https://bupt-yy.github.io/>

June 16, 2021

DISRIPTIVE STATISTICS

基本统计量

- 校正平方和: $CSS = \sum_{j=1}^n w_j (x_j - \bar{x})^2$.
- 未校正平方和: $USS = \sum_{j=1}^n w_j x_j^2$.
- 变异系数 (Coefficient of variation) $CV = \frac{s}{\bar{x}} \times 100\%$
- 偏度系数 $skewness = \frac{n}{(n-1)(n-2)} \sum_{j=1}^n \left(\frac{x_j - \bar{x}}{s} \right)^3$
- 峰度系数 $kurtosis = \frac{n(n+1)}{(n-1)(n-2)(n-3)} \sum_{j=1}^n \left(\frac{x_j - \bar{x}}{s} \right)^4$
- 均值: $\text{mean} = \frac{\sum_{j=1}^n w_j}{n}$, 众数: mode, 中位数 $\text{midian} = P_{50}$.
- 极差: $\text{range} = \max - \min$
- 标准差: $s = \left(\frac{\sum_{j=1}^n w_j (x_j - \bar{x})^2}{n-1} \right)^{1/2}$, 方差: $\text{Var} = s^2$, 标准误差: $\text{std err} = \frac{s}{\sqrt{\text{sumwgt}}}$.
- 权重和: $\text{sum wgt} = \sum_{j=1}^n w_j$, 总和: $\text{sum} = \sum_{j=1}^n w_j x_j$.
- 百分位数 P_1, P_2, \dots, P_{99} , 四分位数 $Q_1 = P_{25}, Q_3 = P_{75}$. 四分位极差 (Interquartile range) $IQR = Q_3 - Q_1$.

```
from scipy import stats
import numpy as np
import math
import matplotlib.pyplot as plt

x = np.random.randn(600)
n = len(x)

print("Descriptive statistics of x")

print("观测个数: %d" % n)
print("校正平方和: %f" % np.sum((x-x.mean())**2, 0)) #CSS
print("非校正平方和: %f" % np.sum(x**2, 0)) #USS
print("变异系数: %f%%" % (stats.variation(x)*math.sqrt(n/(n-1))*100))
```

```

skewness=stats.skew(x, bias=False)
if(skewness<-0.2):
    print("偏度系数为: %f, 数据呈负偏态分布(偏左分布), 相较正态分布, 集中位置偏向数值大的一侧(高峰向右偏移)" %
          skewness)
elif(skewness>0.2):
    print("偏度系数为: %f, 数据呈正偏态分布(偏右分布), 相较正态分布, 集中位置偏向数值小的一侧(高峰向左偏移)" %
          skewness)
else:
    print("偏度系数为: %f, 接近于0, 数据基本呈现对称分布" % skewness)
kurtosis = stats.kurtosis(x, bias=False)
if(kurtosis<-0.2):
    print("峰度系数为: %f, 呈现低峰/薄尾分布(Platykurtic/Light-tailed distr)." %
          kurtosis)
elif(kurtosis>0.2):
    print("峰度系数为: %f, 呈现尖峰/厚尾分布(Leptokurtic/Heavy-tailed distr)." %
          kurtosis)
else:
    print("峰度系数为: %f, 接近于0, 该分布没有尖峰厚尾或峰部平坦的特征" %
          kurtosis)

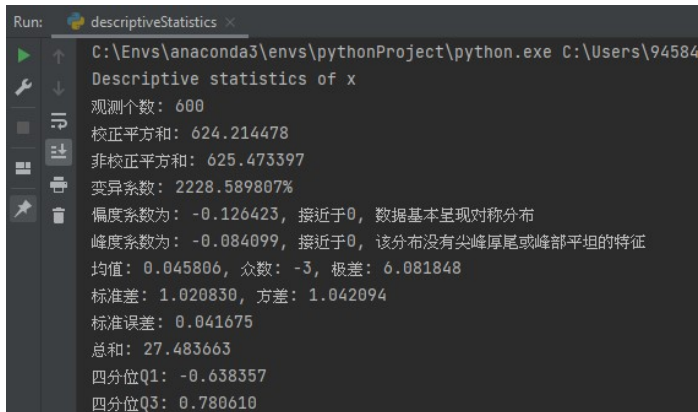
```

```
print("均值: %f" % x.mean(), end=" ")
print("众数: %d" % stats.mode(x)[0], end=" ")
print("极差: %f" % (x.max() - x.min()))
print("标准差: %f" % math.sqrt(stats.tvar(x)), end=" ")
print("方差: %f" % stats.tvar(x))
print("标准误差: %f" % math.sqrt(stats.tvar(x)/n))
print("总和: %f" % np.sum(x))
print("四分位Q1: %f" % np.percentile(x, 25, interpolation='nearest'))
print("四分位Q3: %f" % np.percentile(x, 75, interpolation='nearest'))
```

注: stats.skew() 和 stats.kurtosis 如果不提供参数 bias=False, 则得到未修正的偏度和峰度系数. kurtosis 默认是“减了 3”的峰度系数, 即正态总体的峰度修正为 0.

注: stats.tvar() 方法提供的是无偏的样本方差, 分母是 $n - 1$; 而 stats.var() 提供的是有偏的, 分母是 n .

运行结果:



```
Run: descriptiveStatistics x
C:\Envs\anaconda3\envs\pythonProject\python.exe C:\Users\94584
Descriptive statistics of x
观测个数: 600
校正平方和: 624.214478
非校正平方和: 625.473397
变异系数: 2228.589807%
偏度系数为: -0.126423, 接近于0, 数据基本呈现对称分布
峰度系数为: -0.084099, 接近于0, 该分布没有尖峰厚尾或峰部平坦的特征
均值: 0.045806, 众数: -3, 极差: 6.081848
标准差: 1.020830, 方差: 1.042094
标准误差: 0.041675
总和: 27.483663
四分位Q1: -0.638357
四分位Q3: 0.780610
```

Figure: 1. 基本统计量

与正态分布对比的 Q-Q 图 (Quantile-Quantile plot)

```
def qqplot(data):  
    n = len(data)  
    mu = data.mean()  
    sigma = math.sqrt(stats.tvar(data))  
    normalized_data = (np.sort(data) - mu)/sigma  
  
    t = np.linspace(0.5/n, 1-0.5/n, num=n, endpoint=True)  
    q = stats.norm.ppf(t)  
    fig, ax = plt.subplots()  
    ax.plot(q, normalized_data, label='data')  
    ax.plot(q, q, label='normal distr.')  
    ax.set_xlabel('Expected(Normal Quantiles)')  
    ax.set_ylabel('Observed')  
    ax.set_title('Quantile-Quantile plot')  
    ax.legend()  
    plt.show()
```

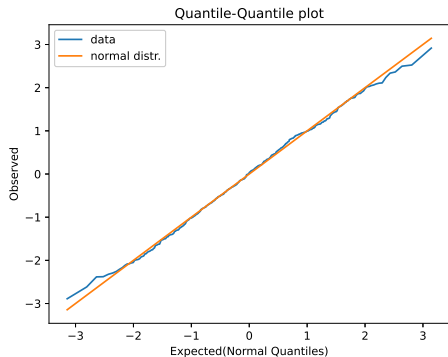


Figure: 2. Q-Q 图

设样本为 x_1, \dots, x_n , 将样本从小到大排列, 得到次序统计量 $x_{(1)}, \dots, x_{(n)}$.

取

$$x_j^* = \frac{x_{(j)} - \bar{x}}{s}, \quad j = 1, \dots, n, \quad (1)$$

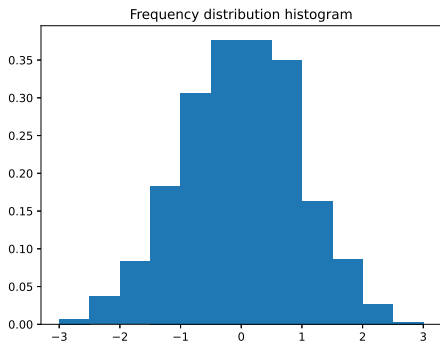
计算正态分布的各个分位数:

$$q_j \Leftarrow \Pr(Z \leq q_j) = \frac{j - 0.5}{n}. \quad (2)$$

在图上标出 (q_j, x_j^*) .

频率分布直方图

```
def histogram(data):  
    # 根据经验公式取合适的区间个数  
    K = int(1 + 4 * math.log(len(data), 10))  
    # 划分一个比较“整”的区间  
    bins = np.linspace(float('% .1g' % data.min()), float('% .1g' % data.max()), K  
                        + 1)  
  
    fig, ax = plt.subplots()  
    ax.set_title('Frequency distribution histogram')  
    plt.hist(data, bins=bins, density=True)  
    plt.show()
```



注: 纵坐标为 $\frac{\text{各段的频率 } f_j}{\text{本段的区间长度}}$

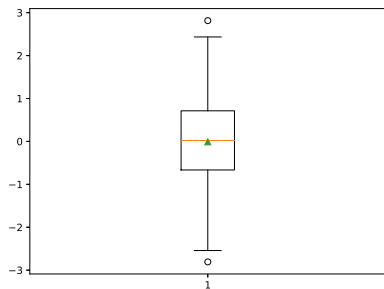
根据经验公式, 划分的区间个数取为

$$K = \lfloor 1 + 4 \lg(n) \rfloor. \quad (3)$$

Figure: 3. 频率分布直方图

箱型图

```
def boxPlot(data):  
    fig, ax = plt.subplots()  
    plt.boxplot(data, showmeans=True)  
    plt.show()
```



最上方的横线表示距离顶部 1.5IQR 的数据点
中间的盒子的三条线分别表示: P_{75} , P_{50} (中位数), P_{25} .
盒子中间的三角表示均值
最下方的横线表示距离底部 1.5IQR 的数据点

Figure: 4. Box Plot

数据的正态性检验.

H_0 : 数据服从正态分布, vs. H_1 : 数据不服从正态分布

拟合优度检验的方法有:

- Kolmogorov-Smirnov Test(D Test)
- Cramer-von Mises Test
- Shapiro-Wilk Test(W Test)
- Anderson-Darling Test

在正态性检验中, 当样本量 $n \leq 2000$ 时, 应该以 Shapiro-Wilk 检验为准; 当样本量 $n > 2000$ 时, 应当以 Kolmogorov-Smirnov 检验为准

```

def fittingTestForNormalDistr(data):
    #标准化
    n = len(data)
    mu, sigma = data.mean(), math.sqrt(stats.tvar(data))
    data = (np.sort(data) - mu)/sigma
    #Shapiro-Wilk检验
    if(n<=5000):
        sw = stats.shapiro(data)
        print("Shapiro-Wilk检验: 统计量W = {0}, P值 = {1}".format(sw[0], sw[1]))
    #Kolmogorov-Smirnov检验
    ks = stats.kstest(data, 'norm')
    print("Kolmogorov-Smirnov 检验: 统计量D = {0}, P值 = {1}".format(ks[0], ks[1]))

    #Cramer-von Mises检验
    cm = stats.cramervonmises(data, 'norm')
    print("Cramer-von Mises 检验: 统计量W-Sq = {0}, P值 = {1}".format(cm.statistic, cm.pvalue))

    #Anderson-Darling检验
    ad = stats.anderson(data, 'norm')[0]

```

```

adstar = ad*(1+0.75/n+2.25/n/n)
adpvalue = 0
if(adstar>=0.6):
    adpvalue = math.exp(1.2937 - 5.709 * adstar + 0.0186 * adstar * adstar)
elif(adstar>0.34):
    adpvalue = math.exp(0.9177 - 4.279 * adstar - 1.38 * adstar * adstar)
elif(adstar>0.2):
    adpvalue = 1-math.exp(-8.318 + 42.796 * adstar - 59.938 * adstar *
                           adstar)
else:
    adpvalue = 1-math.exp(-13.436 + 101.14 * adstar - 223.73 * adstar *
                           adstar)
print("Anderson-Darling检验：统计量A-Sq = {0}, A* = {1}, P值 = {2}".format(
    ad, adstar, adpvalue))

```

```
if(n<=2000):  
    if (sw[1]<0.01):  
        print("结论：该分布高度显著地不符合正态分布")  
    elif (sw[1]<0.05):  
        print("结论：该分布显著不符合正态分布")  
    else:  
        print("结论：没有充足的理由拒绝正态性检验的原假设，应当认为原分布呈  
              正态")  
else:  
    if (ks[1] < 0.01):  
        print("结论：该分布高度显著地不符合正态分布")  
    elif (ks[1] < 0.05):  
        print("结论：该分布显著不符合正态分布")  
    else:  
        print("结论：没有充足的理由拒绝正态性检验的原假设，应当认为原分布呈  
              正态")
```

正态性检验运行结果:

```
Shapiro-Wilk检验: 统计量W = 0.9976682662963867, P值 = 0.5752055644989014  
Kolmogorov-Smirnov 检验: 统计量D = 0.025628664719309913, P值 = 0.8157104154736606  
Cramer-von Mises 检验: 统计量W-Sq = 0.05259967887435022, P值 = 0.8605205149449479  
Anderson-Darling检验: 统计量A-Sq = 0.34939167318111686, A* = 0.34983059647055065, P值 = 0.47325945127643976  
结论: 没有充足的理由拒绝正态性检验的原假设, 应当认为原分布呈正态
```

Figure: 5. 正态性检验

经验分布函数 (EDF, Empirical distr. func.): 设样本为 X_1, \dots, X_n .

$$F_n(x) = \begin{cases} 0, & x < X_{(1)}, \\ \frac{j}{n}, & X_{(j)} \leq x < X_{(j+1)}, j = 1, \dots, n \\ 1, & X_{(n)} \leq x. \end{cases} \quad (4)$$

Kolmogorov-Smirnov Test Statistic(D):

$$D := \sup_x |F_n(x) - F(x)|. \quad (5)$$

$$D^+ = \max_j \left(\frac{j}{n} - F(X_{(j)}) \right)$$

$$D^- = \max_j \left(F(X_{(j)}) - \frac{j-1}{n} \right)$$

$$D = \max(D^+, D^-).$$

Anderson-Darling Statistic:

$$\begin{aligned} A^2 &= n \int_{-\infty}^{+\infty} \frac{(F_n(x) - F(x))^2}{F(x)(1 - F(x))} dF(x) \\ &= -n - \frac{1}{n} \sum_{j=1}^n [(2j-1) \log(F(X_{(j)})) + (2n+1-2j) \log(1 - F(X_{(j)}))] \end{aligned} \quad (6)$$

Cramér-von Mises statistic:

$$\begin{aligned} W^2 &= n \int_{-\infty}^{+\infty} (F_n(x) - F(x))^2 dF(x) \\ &= \sum_{j=1}^n \left(F(X_{(j)}) - \frac{2j-1}{2n} \right)^2 + \frac{1}{12n}. \end{aligned} \quad (7)$$

Shapiro-Wilk Statistic:

$$W = \frac{\left(\sum_{j=1}^n (a_j x_{(j)}) \right)^2}{\sum_{j=1}^n (x_j - \bar{x})^2}. \quad (8)$$

其中 $(a_1, \dots, a_n) = \frac{m^\top V^{-1}}{\|V^{-1}m\|}$. 这里 $m = (m_1, \dots, m_n)^\top$. m 和 V 分别是服从标准正态分布的简单随机样本的次序统计量的期望和方差.

LOGISTIC 回归模型

sigmoid 函数:

$$\sigma(x) = \frac{1}{1 + e^{-x}}. \quad (9)$$

sigmoid 函数可将任一实数映上 $(0, 1)$. 因此, 逻辑回归可作为任何一个分类的后验概率:

$$\begin{aligned} P(C = 1|x) &= y(x) = \sigma(w^T x + b), \\ P(C = 0|x) &= 1 - y. \end{aligned} \quad (10)$$

这两个公式整合在一起就是

$$P(C = t|x) = y^t(1 - y)^{1-t}, \quad t \in \{0, 1\}. \quad (11)$$

似然函数:

$$L(w, b) = \prod_{j=1}^n y_j^{t_j} (1 - y_j)^{1-t_j}, y_j = P(C = 1|x_j). \quad (12)$$

求最大似然估计就是最小化:

$$E = -\ln L(w, b) = -\sum_{j=1}^n [t_j \ln y_j + (1 - t_j) \ln(1 - y_j)]. \quad (13)$$

这是 Cross-Entropy Error func.

Gradient Descent:

$$\begin{aligned}w^{(k+1)} &= w^{(k)} - \eta \frac{\partial E}{\partial w} = w^{(k)} + \eta \sum_{j=1}^n (t_j - y_j) x_j \\b^{(k+1)} &= b^{(k)} - \eta \frac{\partial E}{\partial b} = b^{(k)} + \eta \sum_{j=1}^n (t_j - y_j)\end{aligned}\tag{14}$$

问题: 数据集变大后, 求和的计算开销也迅速增加.

解决: 从数据集中随机选取部分数据, 进行在线迭代 (随机梯度下降, SGD). 每次参数刷新使用的子集叫小批量 (Mini-Batch), 使用小批量的 SGD 叫做 MSGD.

多分类 LOGISTIC 回归模型

softmax 函数:

$$P(C = k|x) = y_k(x) = \frac{\exp(w_k^\top x + b_k)}{\sum_{k=1}^K \exp(w_k^\top x + b_k)}. \quad (15)$$

maximum likelihood:

$$E = -\ln L(W, b) = -\sum_{j=1}^n \sum_{k=1}^K t_{jk} \ln(y_{jk}). \quad (16)$$

这里 $W = [w_1, \dots, w_K]$, $b = [b_1, \dots, b_K]$, $y_{jk} = y_k(x_j)$. 如果第 j 个输入数据 x_j 属于类 k , 则 $t_{jk} = 1$, 否则 $t_{jk} = 0$.

$$\begin{aligned} \frac{\partial E}{\partial w_k} &= -\sum_{j=1}^n (t_{jk} - y_{jk}) x_j \\ \frac{\partial E}{\partial b_k} &= -\sum_{j=1}^n (t_{jk} - y_{jk}) \end{aligned} \quad (17)$$

Logistic 回归的适用条件:

- 数据来自随机样本
- 因变量被假设为若干自变量的函数, 且是非线性的
- Logistic 对共线性敏感, 当自变量之间存在高度自相关时, 会导致估计的标准误差膨胀
- 不要求残差是独立同分布的.
- Logistic 没有关于自变量分布的假设条件, 自变量可以是连续变量、分类变量等.

Logistic 的极大似然估计具有的性质:

- 相合性 (consistent)
- 渐进有效性 (asymptotically efficient)
- 渐进正态性 (asymptotically normal)

可以用渐进正态性进行参数的显著性检验、计算置信区间等等.

样本多大时, 可以用 Logistic 的 MLE?

这个问题至今没有明确的答案. 较普遍的看法: 中等规模样本 ($n = 100$) 可接受, 当样本量 $n < 100$ 时使用这个 MLE 对概率值进行估计和计算置信区间会带来较大的风险. 若样本量 $n > 500$, 使用这个 MLE 就比较充分了.

当有许多参数需要估计/自变量间有高度的共线性时, 需要更大的样本量.

拟合优度的评价

- AIC 准则 (Akaike's A Information Criterion)

$$\text{AIC} = -2 \log(L) + 2K. \quad (18)$$

- SBC 准则 (Schwarz's Bayesian Information Criterion)

$$\text{SBC} = -2 \log(L) + K \log(n), \quad (19)$$

其中 L 为似然函数的取值, K 为参数的个数, n 为样本容量.
AIC/SBC 取值越小说明模型拟合得越好.

预测准确性的评价指标:

- $\text{Gamma} = \frac{nc - nd}{nc + nd},$
- $\text{Somer's D} = \frac{nc - nd}{t},$
- $\text{Tau - a} = \frac{nc - nd}{0.5n(n-1)},$
- $c = \frac{nc + 0.5(t - nc - nd)}{t}.$

其中, n 为样本容量, t 为总的观测个数, nc 是和谐对的数量, nd 是不和谐对的数量.

THANKS