基于统计计算的工业机械设备故障分析

林世铮 李岳锴 李毓晟 林瑞奇

摘　要

机械设备运转过程中会产生不可避免的磨损、老化等问题。随着损耗的增加，会发生各种不同类型的故障，影响生产质量和效率。实际生产中，若能根据机械设备的使用情况，提前预测潜在的故障风险，精准地进行检修维护，维持机械设备稳定运转，不但能够确保整体工业环境运行具备稳定性，也能切实帮助企业提高经济效益。

本项目使用某企业机械设备的使用情况及故障发生情况的数据集，基于**参数估计、假设检验、Bootstrap 重抽样**等统计计算方法，对机械设备的故障情况进行统计分析，最终给出了机械不同故障的可能成因。

在对非故障机器正常运行时的参数分析时，首先我们使用**相关系数检验**，简单探究了各个连续型参数的内部相关性，得到了**室温与机器温度呈正相关、转速与扭矩呈负相关**的结论。其次，我们基于样本数据，对各个连续型参数的**期望、标准差、偏度与峰度**进行估计，并使用 **Bootstrap 重抽样方法**，对样本估计量的偏差与标准差作了数值估计，给出了**基本 Bootstrap 置信区间**。最后，我们调用 Python 的第三方库 Fitter，遍历各种概率分布对各个连续型参数的分布作拟合，最终确定了各项参数的**近似分布**。

以非故障类机器的参数分布为基础，我们对各种故障机器的参数进行统计分析。首先我们针对机器质量等级这一定序特征，绘制了不同类别机器的质量等级饼图，并通过**列联分析检验**定量检验故障与机器质量等级的关系，得到结论**质量等级较低的机器容易发生 OSF 类过载故障**。随后，我们构造了 **K-S 检验**，依次检验故障类设备的连续型特征是否与非故障类设备同分布，结合直方图探究各故障的成因，最终给出了发生各类故障可能的原因。

本项目综合使用了多种统计计算方法，从不同角度对机械设备故障情况进行统计分析，具有很好的现实意义与实际应用价值。

关键词：统计计算；机械故障分析；拟合优度检验；参数估计；重抽样方法

目录

第 1 章　引言

1.1　课题背景

制造业是国民经济的主体，近十年来，嫦娥探月、祝融探火、北斗组网，一大批重大标志性创新成果引领中国制造业不断攀上新高度。作为制造业的核心，机械设备在工业生产的各个环节都扮演着不可或缺的重要角色。

图 1: 工业 4.0 蓝图

在工业 4.0 的大背景下，大型机械设备的结构愈发复杂，相应的工作原理与故障机理也随之复杂化与综合化，使得设备故障的预警与诊断难度逐渐增大。关键设备一旦发生故障，可能引发一系列连锁反应，进而影响整个生产系统，造成大面积停产，带来巨大经济损失。此外，机械设备由于所处生产环境的复杂性，若发生事故可能有爆炸、火灾、有毒气体液体泄漏等诸多风险这种风险不仅严重危及工作人员与厂区周围居民的人身安全，甚至可能造成永久性的生态环境污染。因此，怎样实现准确及时的故障预警与诊断，成为了众多国内外行业专家、学者的重点研究对象。

1.2　研究目的

机械设备运转过程中会产生不可避免的磨损、老化等问题。随着损耗的增加，会发生各种**不同类型的故障**，影响生产质量和效率。实际生产中，若能根据机械设备的使用情况，提前预测潜在的故障风险，精准地进行检修维护，维持机械设备稳定运转，不但能够确保整体工业环境运行具备稳定性，也能切实帮助企业提高经济效益。

随着机械产业和机械设备趋于复杂化、大型化，依托于专家经验选取合适的运行参数特征并据此进行预警与诊断的方式，显然已经难以适应设备复杂的故障机理。在这样的背景下，工业互联网的新议题驱动了机械设备大数据时代的诞生，分布式架构、云计算、人工智能等新兴 IT 技术的发展，使得机械设备长期采集的海量运行数据得以有效处理，**基于数据驱动的机械设备故障预警与诊断方法**也有了落地的土壤，并逐步成为该领域的热点研究对象。

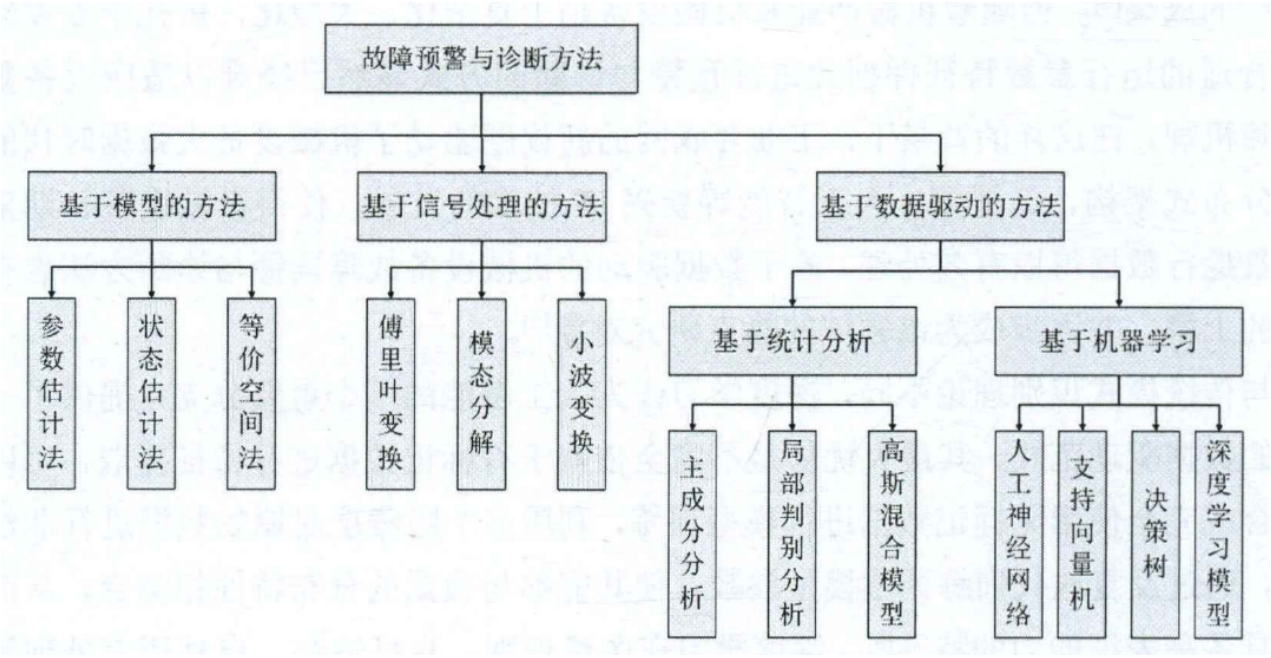

图 2: 机械故障预警与诊断方法

在本项目中，我们将寻找并构造合适的机械设备运行参数数据，基于统计计算的相关方法进行分析。首先，我们提取非故障机械设备的数据，对其参数指标的统计分布特征进行估计与拟合。在对正常运行的机械设备参数进行统计分析后，对于不同类型的故障设备，我们将探究**每类故障的主要成因**，找出与其相关的特征属性，进行量化分析，挖掘可能存在的模式或规则。

1.3　数据集说明

我们从互联网中爬取了某企业机械设备的使用情况及故障发生情况的数据集（见“train_data.xlsx”），用于设备故障分析及故障主要相关因素的探究。经过一些基本数据预处理后，得到的数据集包含 8992 行，每一行数据记录了机械设备对应的运转及故障发生情况记录。

机器编号	统一规范代码	机器质量等级	室温（K）	机器温度（K）
84	L48027	L	296.4	307.4
6986	M15740	M	295.8	306.3
8047	L48083	L	295.7	306.2
4425	M15847	M	296.3	307.1
4519	H30402	H	296.3	307.1

转速（rpm）	扭矩（Nm）	使用时长（min）	是否发生故障	具体故障类别
2833	5.6	213	1	PWF
1235	76.2	89	1	PWF
2270	14.6	149	1	PWF
1534	33.8	151	0	Normal
1774	25.9	154	0	Normal

表 1: 数据集示例

数据提供了实际生产中常见的机械设备使用环境和工作强度等指标，包含不同设备所处厂房的室温（单位为开尔文 K），其工作时的机器温度（单位为开尔文 K）、转速（单位为每分钟的旋转次数 rpm）、扭矩（单位为牛米 Nm）及机器运转时长（单位为分钟 min）。除此之外，数据集还包括了机械设备的统一规范代码、质量等级及在该企业中的机器编号，其中质量等级分为高、中、低（H、M、L）三个等级。对于机械设备的故障情况，数据提供了两列数据描述——“是否发生故障”和“具体故障类别”。其中“是否发生故障”取值为 0/1，0 代表设备正常运转，1 代表设备发生故障；“具体故障类别”包含五种情况，分别是 Normal、TWF、HDF、PWF、OSF，其中，Normal 代表设备正常运转（与“是否发生故障”为 0 相对应），其余代码代表的是发生故障的类别，包含 5 种，其中 TWF 代表磨损故障，HDF 代表散热故障，PWF 代表电力故障，OSF 代表过载故障。

由于机械设备在使用环境以及工作强度上存在较大差异，因此，我们需要针对故障与非故障的机械设备，以及不同的故障类型分别进行分析。

第 2 章　非故障类设备的统计规律

2.1　相关系数检验

在数据集中，共有五种连续取值的特征。为了后续对非故障设备参数的统计分析，我们首先探究各个连续型特征的相关性。对于两个连续型变量而言，常用相关系数来衡量二者间的相关程度。首先考虑两个变量间的线性相关关系．变量 X 和 Y 间的相关系数定义为 ρ, 即:

$$\rho = \frac{E(XY) - E(X)E(Y)}{\sqrt{\text{Var}(X)\,\text{Var}(Y)}} \tag{1}$$

显然，ρ 取值在-1 到 1 之间。$|\rho| = 1$ 说明 X 与 Y 之间存在完全线性关系。如果 X 和 Y 相互独立，则有 $\rho = 0$. 关心两个变量间是否存在简单线性关系，即检验：$H_0 : \rho = 0, H_1 : \rho \neq 0$

样本 $(X_1, Y_1), (X_2, Y_2), \cdots, (X_n, Y_n)$ 的相关程度用 Pearson 相关系数 r 度量，即：

$$r = \frac{\sum_{i=1}^{n} \left(X_i - \bar{X}\right)\left(Y_i - \bar{Y}\right)}{\sqrt{\sum_{i=1}^{n} \left(X_i - \bar{X}\right)^2 \sum_{i=1}^{n} \left(Y_i - \bar{Y}\right)^2}} \tag{2}$$

当 (X, Y) 服从二元正态分布时，在原假设下，可以证明：

$$t = r\sqrt{\frac{n-2}{1-r^2}} \sim t(n-2) \tag{3}$$

我们通过编程计算室温、机器温度、转速、扭矩与使用时长的 Pearson 相关系数。各个相关系数的显著性水平如下表所示：

	室温（K）	机器温度（K）	转速（rpm）	扭矩（Nm）	使用时长（min）
室温（K）		0	0	0	0.048
机器温度（K）	0		0.014	0.025	0.142
转速（rpm）	0	0.014		0	0.05
扭矩（Nm）	0	0.025	0		0.007
使用时长（min）	0.048	0.05	0.007	0	

表 2: 相关系数的显著性水平

相关系数及其热力图如下图所示：

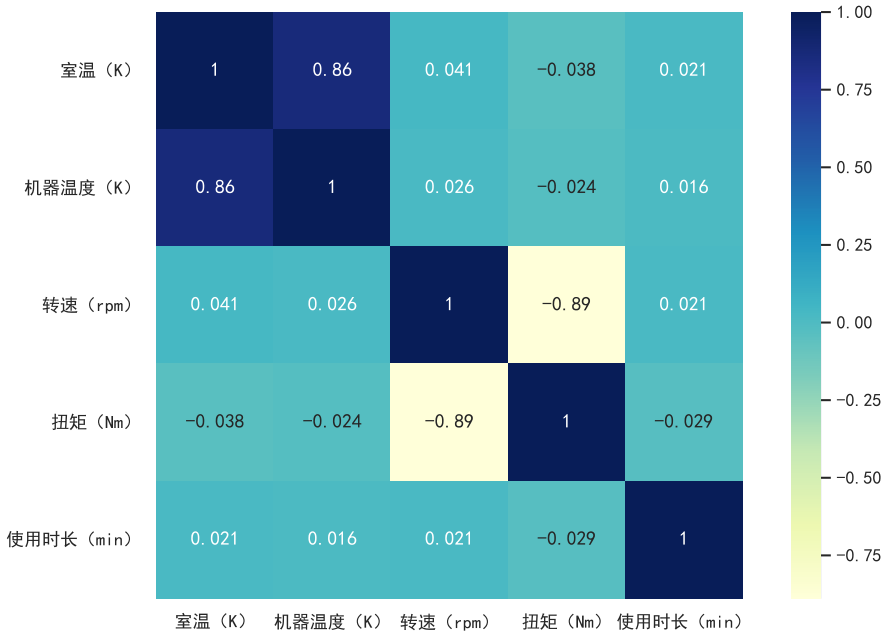

图 3: 连续特征的相关系数热力图

从热力图中不难看出，**室温与机器温度之间存在显著的正相关性，而转速与扭矩之间存在显著的负相关性**。这两组特征的相关性将作为先验的统计规律，为后续的探究提供前置信息。

2.2　矩统计量的数值估计

为了得到机器正常运作时各参数的分布情况，我们需要对各个参数的矩统计量进行数值估计。通过样本计算得到总体期望的估计量之后，我们希望探究该估计的偏差与标准差，并给出一个估计量的置信区间。由于总体样本的分布未知，我们选择**基于 Bootstrap 的重抽样方法**，对估计量的偏差、标准差以及置信区间进行数值计算。Bootstrap 方法的基本步骤如下：

算法 1 Bootstrap 方法

输入： 样本 $X_1, X_2, \cdots X_n$

输出： 估计量 $\hat{\theta}$ 分布特征的近似估计 $\widehat{Bias}\left(\hat{\theta}\right)$, $\widehat{SE}\left(\hat{\theta}\right)$，以及基本 Bootstrap 置信区间

1: 计算估计量 $\hat{\theta} = \hat{\theta}(X_1, X_2, \cdots X_n)$
2: **for** $i = 1, 2, \cdots, m$ **do**
3: 　从样本 $X_1, X_2, \cdots X_n$ 中有放回地产生数据 $X_{i1}^*, X_{i2}^*, \cdots X_{in}^*$
4: 　得到 θ 的估计 $\hat{\theta}_i^* = \hat{\theta}_i^*\left(X_{i1}^*, X_{i2}^*, \cdots X_{in}^*\right)$
5: **end for**
6: 计算 $\widehat{Bias}\left(\hat{\theta}\right) = \hat{E}\left(\hat{\theta}\right) - \hat{\theta} = \dfrac{\sum_{i=1}^{m}\hat{\theta}_i^*}{m} - \hat{\theta}$
7: 计算 $\widehat{SE}\left(\hat{\theta}\right) = \sqrt{\dfrac{\sum_{i=1}^{m}\left(\hat{\theta}_i^* - \bar{\hat{\theta}}^*\right)^2}{m}}$
8: 计算基本 Bootstrap 置信区间：$\left[\hat{\theta}^*_{\left(\left[\frac{m\alpha}{2}\right]\right)},\ \hat{\theta}^*_{\left(\left[\frac{m(1-\alpha)}{2}\right]\right)}\right]$
9: **return** $\widehat{Bias}\left(\hat{\theta}\right)$, $\widehat{SE}\left(\hat{\theta}\right)$，基本 Bootstrap 置信区间

基于以上算法流程，下面我们依次给出期望、标准差、偏度与峰度的数值估计。

2.2.1 期望的数值估计

在统计计算中，我们通常使用样本均值 $\bar{X}$，作为总体期望 μ 的估计值。样本均值的表达式如下：

$$\bar{X} = \frac{\sum_{i=1}^{n} X_i}{n} \tag{4}$$

显然，样本均值是总体期望的**无偏估计量**，即：

$$E\left[\bar{X}\right] = \mu \tag{5}$$

基于前文所述的算法流程，我们通过编程计算了样本均值，并使用 Bootstrap 方法估计了样本均值的偏差、标准差以及置信区间如下：

参数	样本均值 $\bar{X}$	偏差估计 $\widehat{Bias}\left(\bar{X}\right)$	标准差估计 $\widehat{SE}\left(\bar{X}\right)$	置信区间
室温（K）	300.247	-1.4×10^{-5}	1.907×10^{-2}	[300.209, 300.283]
机器温度（K）	310.176	4.0×10^{-6}	1.479×10^{-2}	[310.146, 310.204]
转速（rpm）	1540.884	-7.401×10^{-2}	1.663	[1537.685 ,1544.431]
扭矩（Nm）	39.623	1.168×10^{-3}	9.546×10^{-2}	[39.429, 39.804]
使用时长（min）	106.610	-4.089×10^{-2}	6.452×10^{-1}	[105.502, 107.906]

表 3: 均值的数值估计

2.2.2　标准差的数值估计

在本项目中，我们使用样本标准差 $\hat{\sigma}$，作为总体标准差 σ 的估计值。这里的样本标准差的表达式如下：

$$\hat{\sigma} = \sqrt{\frac{\sum_{i=1}^{n}\left(X_i - \bar{X}\right)^2}{n}} \tag{6}$$

由统计学基础知识可知，$\hat{\sigma}$ 是 σ 的**有偏估计量**，且在小样本条件下，有 $\hat{\sigma} < \sigma$.

基于前文所述的算法流程，我们通过编程计算了样本标准差，并使用 Bootstrap 方法估计了样本均值的偏差、标准差以及置信区间如下：

参数	样本标准差 $\hat{\sigma}$	偏差估计 $\widehat{Bias}\left(\hat{\sigma}\right)$	标准差估计 $\widehat{SE}\left(\hat{\sigma}\right)$	置信区间
室温（K）	3.561	-5.430×10^{-4}	3.989×10^{-2}	[3.4805,3.638]
机器温度（K）	2.020	-3.1×10^{-4}	2.468×10^{-2}	[1.973, 2.071]
转速（rpm）	28332.043	-48.031	743.785	[26858.489, 29900.599]
扭矩（Nm）	90.632	7.275×10^{-2}	1.279	[88.189, 93.142]
使用时长（min）	3956.497	-1.351×10^{-1}	39.572	[3881.894, 4038.741]

表 4: 标准差的数值估计

Bootstrap 估计的偏差均为负值，恰好验证了小样本条件下 $\hat{\sigma} < \sigma$ 这一关系，说明我们的估计方法是正确有效的。

2.2.3　偏度的数值估计

偏度（skewness），是统计数据分布偏斜方向和程度的度量，是统计数据分布非对称程度的数字特征，表征概率分布密度曲线相对于平均值不对称程度的特征数。

偏度是样本的三阶标准化矩，定义式如下：

$$\text{Skew}(X) = E\left[\left(\frac{X-\mu}{\sigma}\right)^3\right] = \frac{k_3}{\sigma^3} = \frac{k_3}{k_2^{3/2}} \tag{7}$$

其中 k_2, k_3 分别表示二阶和三阶中心矩。我们直接使用 k_2, k_3 的样本估计 $\hat{k_2}, \hat{k_3}$，对样本的偏度进行估计如下：

$$\widehat{Skew}\,(X) = \frac{\hat{k}_3}{\hat{k}_2^{3/2}} \tag{8}$$

基于前文所述的算法流程，我们通过编程计算了样本偏度，并使用 Bootstrap 方法估计了样本偏度的偏差、标准差以及置信区间如下：

参数	样本偏度 $\widehat{Skew}\,(X)$	偏差估计 $\widehat{Bias}\left(\widehat{Skew}\,(X)\right)$	标准差估计 $\widehat{SE}\left(\widehat{Skew}\,(X)\right)$
室温（K）	9.678×10^{-2}	9.010×10^{-4}	1.539×10^{-2}
机器温度（K）	3.880×10^{-2}	8.780×10^{-4}	1.633×10^{-2}
转速（rpm）	1.602	-2.434×10^{-3}	5.500×10^{-2}
扭矩（Nm）	-5.806×10^{-2}	9.29×10^{-4}	2.027×10^{-2}
使用时长（min）	2.829×10^{-2}	7.74×10^{-4}	1.487×10^{-2}

表 5: 偏度的数值估计

2.2.4　峰度的数值估计

峰度（kurtosis）又称峰态系数，表征概率密度分布曲线在平均值处峰值高低的特征数。在统计学中，峰度衡量实数随机变量概率分布的峰态。峰度高就意味着方差增大是由低频度的大于或小于平均值的极端差值引起的。

峰度被定义为四阶累积量除以二阶累积量的平方，它等于四阶中心矩除以概率分布方差的平方再减去 3：

$$Kurt\,(X) = \frac{k_4}{k_2^2} - 3 \tag{9}$$

类似地，我们直接使用 k_2, k_4 的样本估计 $\hat{k_2}, \hat{k_4}$，对样本的峰度进行估计如下：

$$\widehat{Kurt}\,(X) = \frac{\hat{k}_4}{\hat{k}_2^2} - 3 \tag{10}$$

基于前文所述的算法流程，我们通过编程计算了样本峰度，并使用 Bootstrap 方法估计了样本峰度的偏差、标准差以及置信区间如下：

参数	样本峰度 $\widehat{Kurt}(X)$	偏差估计 $\widehat{Bias}\left(\widehat{Kurt}(X)\right)$	标准差估计 $\widehat{SE}\left(\widehat{Kurt}(X)\right)$
室温（K）	-0.8155	-4.68×10^{-4}	1.774×10^{-2}
机器温度（K）	-0.6479	-2.19×10^{-4}	2.129×10^{-2}
转速（rpm）	4.415	-9.092×10^{-3}	0.3471
扭矩（Nm）	-0.2723	4.5×10^{-5}	2.965×10^{-2}
使用时长（min）	-1.1684	1.64×10^{-4}	1.282×10^{-2}

表 6: 峰度的数值估计

2.3　非故障类特征的拟合分布

对于一组样本，Python 的第三方库 fitter 常常用于初步预测其可能服从的分布。在 Scipy 库中内置了八十多个概率分布，fitter 库可以遍历这些分布，借助偏移参数 loc 与缩放参数 scale 调整随机变量取值范围，得到特定指标下与样本分布最接近的概率分布及其相应的参数取值。

例如，假定拟合得到的最优分布为标准正态分布 norm，偏移参数 $loc = 1$，$scale = 2$。设随机变量 $X \sim N(0,1)$，则样本所在的总体分布 Y 近似服从分布 $scale * X + loc$，即 $Y \sim 2X + 1$。

对于每个分布函数的拟合结果，fitter 库内置了 K-S 检验方法，可以对每个拟合分布的显著性进行评估。单样本情形下，K-S 检验用于检验观测样本的经验分布函数与假设理论分布函数是否一致。设 $X_1, X_2, \cdots, X_n$ 是 n 个独立观测样本，该问题的原假设为 H_0：该样本来自分布函数为 $F(\cdot)$ 的总体。

设 $X_{(1)} \leq X_{(2)} \leq \cdots \leq X_{(n)}$ 为次序样本，经验分布函数为 $F_n\left(X_{(i)}\right) = \dfrac{i}{n}, i = 1, 2, \cdots n$. 经验分布函数与原假设下的分布函数之间的最大偏差记为 K-S 统计量，即：

$$K_n = \max_{1 \leq i \leq n} \left[\max \left(\left| F\left(X_{(i)}\right) - \frac{i}{n} \right|, \left| F\left(X_{(i)}\right) - \frac{i-1}{n} \right| \right) \right] \tag{11}$$

H_0 成立时，K-S 统计量依分布收敛于 Kolmogorov 分布，其分布函数为：

$$P\left(K_n \leq x\right) = 1 - 2\sum_{j=1}^{\infty} (-1)^j \exp\left(-2nj^2x^2\right), \quad 0 < x < +\infty \tag{12}$$

对于给定的显著性水平 α, 当 K_N 大于临界值时，拒绝 H_0。

下面，我们对室温、机器温度、转速、扭矩与使用时长依次使用 fitter 库拟合。

2.3.1　室温分布的拟合

经编程计算，得到拟合效果较好的分布如下：

分布	SSE	K-S 统计量	p 值
johnsonsb	0.5042	0.03350	6.496×10^{-9}
beta	0.5054	0.03608	2.837×10^{-10}
anglit	0.5081	0.04062	6.619×10^{-13}
exponweib	0.5087	0.03720	6.819×10^{-11}
exponpow	0.5094	0.03727	6.231×10^{-11}

表 7: 室温分布的拟合结果

其中，johnsonsb 代表 Johnson SB 分布，beta 代表 Beta 分布，exponweib 代表指数韦布尔分布，exponpow 代表指数幂分布；SSE 刻画了样本值与拟合分布的误差平方和。

由于 Beta 分布较为常见，且 K-S 检验 p 值相对较大，因此我们选取 **Beta 分布**，作为非故障类机器室温的近似分布。

Beta 分布是一组定义在区间 $(0,1)$ 上的连续概率分布，有两个参数 $a>0, b>0$。设随机变量 X 服从参数为 a, b 的 Beta 分布，记作 $X \sim B(a,b)$，那么 X 具有如下概率密度函数：

$$f(x;a,b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} x^{a-1}(1-x)^{b-1}, 0<x<1 \tag{13}$$

其中 Γ 为伽马函数。

Beta 分布的拟合效果如下图所示：

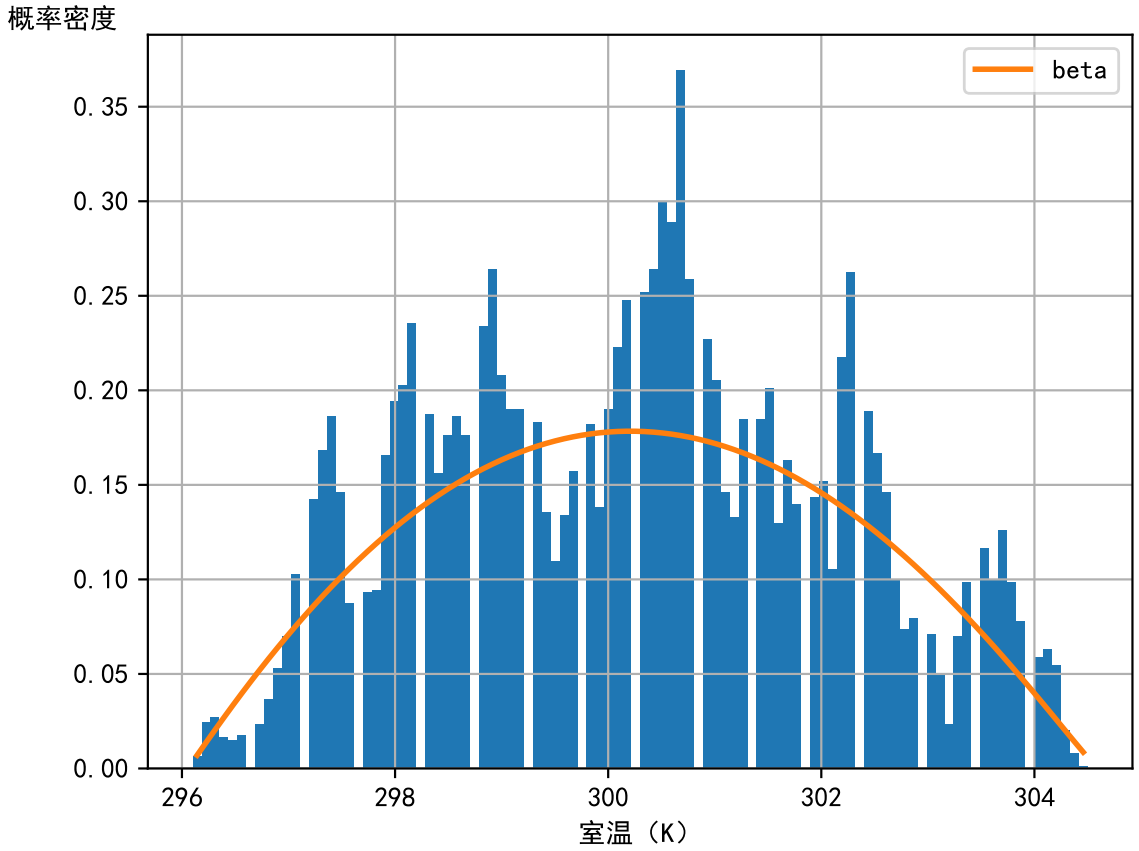

图 4: 室温的近似分布直方图

其中各个参数取值分别为 $a = 2.027$，$b = 2.086$，$loc = 296.053$，$scale = 8.543$。

2.3.2　机器温度分布的拟合

经编程计算，得到拟合效果较好的分布如下：

分布	SSE	K-S 统计量	p 值
cosine	1.482	0.04014	1.300×10^{-12}
gennorm	1.489	0.03458	1.806×10^{-9}
beta	1.490	0.04666	6.875×10^{-17}
frechet_r	1.501	0.05376	2.775×10^{-22}
norm	1.507	0.04649	8.996×10^{-17}

表 8: 机器温度分布的拟合结果

其中，cosine 代表余弦分布，gennorm 代表广义正态分布.

考虑到机器温度与室温存在一定程度的正相关性，结合拟合结果，我们仍然选择**Beta 分布**作为的机器温度的近似分布。拟合效果如下图所示：

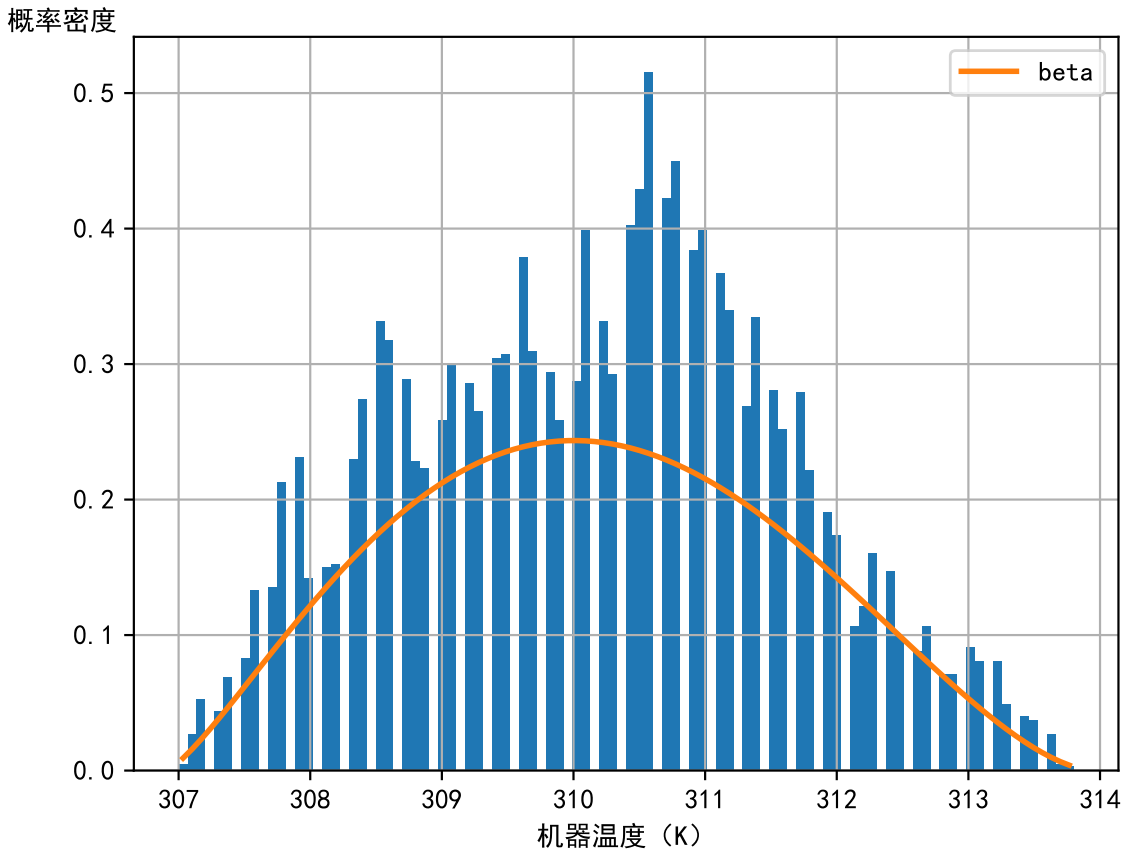

图 5: 机器温度的近似分布直方图

其中各个参数取值分别为 $a = 2.346$，$b = 2.716$，$loc = 306.901$，$scale = 7.058$。

2.3.3　转速分布的拟合

经编程计算，得到拟合效果较好的分布如下：

分布	SSE	K-S 统计量	p 值
gumbel_r	1×10^{-6}	0.02083	1.037×10^{-3}
moyal	2×10^{-6}	0.02426	7.313×10^{-5}
kstwobign	3×10^{-6}	0.04400	4.526×10^{-15}
hypsecant	7×10^{-6}	0.06581	3.447×10^{-33}
dweibull	8×10^{-6}	0.07122	8.255×10^{-39}

表 9: 转速分布的拟合结果

其中，gumbel_r 代表右拖尾耿贝尔分布，moyal 代表 Moyal 分布，hypsecant 代表双曲线正割分布，dweibull 代表双韦布尔分布。

我们选取 K-S 检验 p 值最大的**右拖尾耿贝尔分布**，作为非故障类机器转速的近似分布。

耿贝尔分布是一组定义在区间 $(-\infty, \infty)$ 上的连续概率分布。有中心参数 μ 和展宽参数 σ 两个参数。设随机变量 X 服从参数为 μ, σ 的耿贝尔分布，考虑广义极值分布的概率分布函数：

$$f(x; a, b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} x^{a-1}(1-x)^{b-1}, 0 < x < 1 \tag{14}$$

其中 k 为形状参数，则 $\lim_{k \to 0} F(x; k, \mu, \sigma)$ 即为 X 的概率分布函数：

$$F(x; \mu, \sigma) = \exp\left(-\exp\left(-\frac{x-\mu}{\sigma}\right)\right), \ -\infty < x < \infty \tag{15}$$

将中心参数 $\mu = 0$ 和展宽参数 $\sigma = 1$ 的耿贝尔分布作右拖尾处理，就得到了标准右拖尾耿贝尔分布的概率分布函数：

$$F(x) = \exp\left(-\left(x + \exp\left(-x\right)\right)\right), \ -\infty < x < \infty \tag{16}$$

右拖尾耿贝尔分布的拟合效果如下图所示：

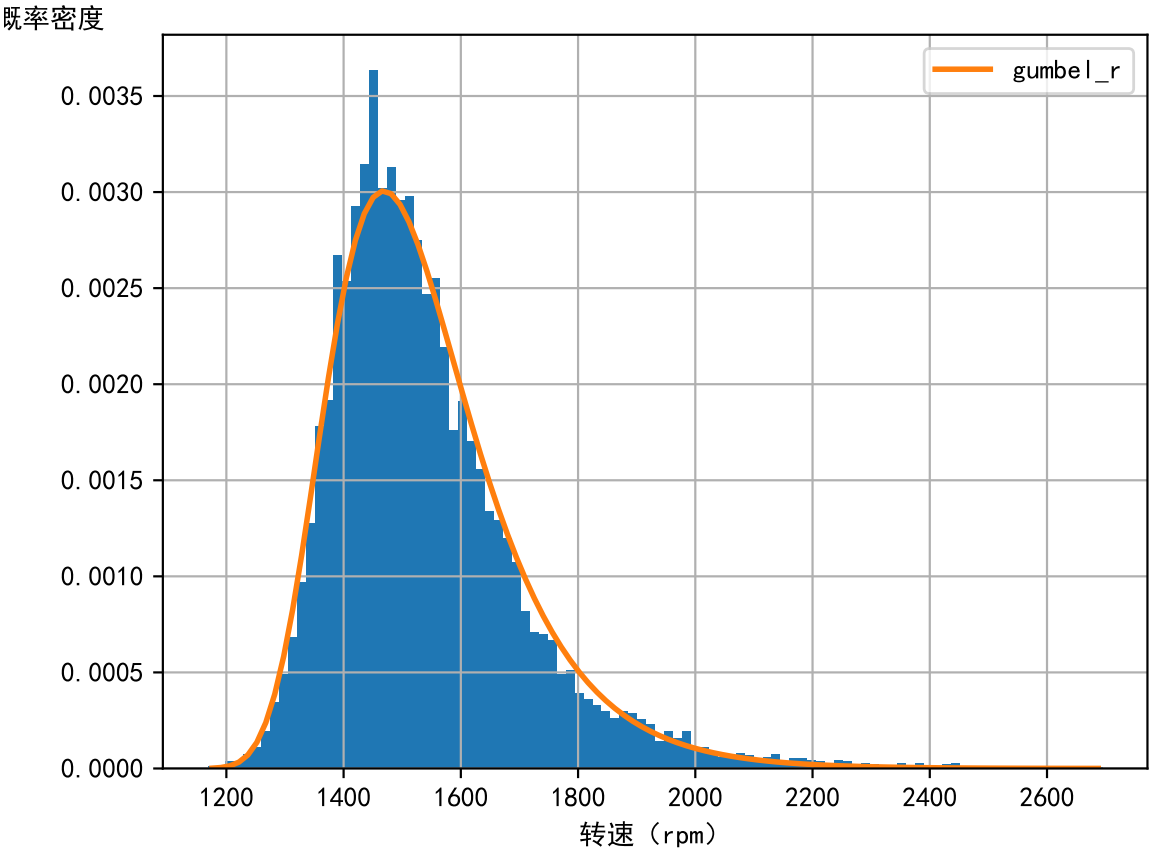

图 6: 转速的近似分布直方图

其中各个参数取值分别为 $loc = 1468.490$，$scale = 122.356$。

2.3.4　扭矩分布的拟合

经编程计算，得到拟合效果较好的分布如下：

分布	SSE	K-S 统计量	p 值
norm	0.0006850	0.01374	7.414×10^{-2}
logistic	0.001142	0.02336	1.483×10^{-4}
genlogistic	0.001144	0.02327	1.598×10^{-4}
hypsecant	0.001732	0.03156	5.829×10^{-8}
gumbel_l	0.002444	0.05287	1.433×10^{-21}

表 10: 扭矩分布的拟合结果

其中，logistic 代表 Logistic 分布，genlogistic 代表广义 Logistic 分布，gumbel_l 代表左拖尾耿贝尔分布。

我们选取 K-S 检验 p 值最大的**正态分布**，作为非故障类机器扭矩的近似分布。

正态分布的拟合效果如下图所示：

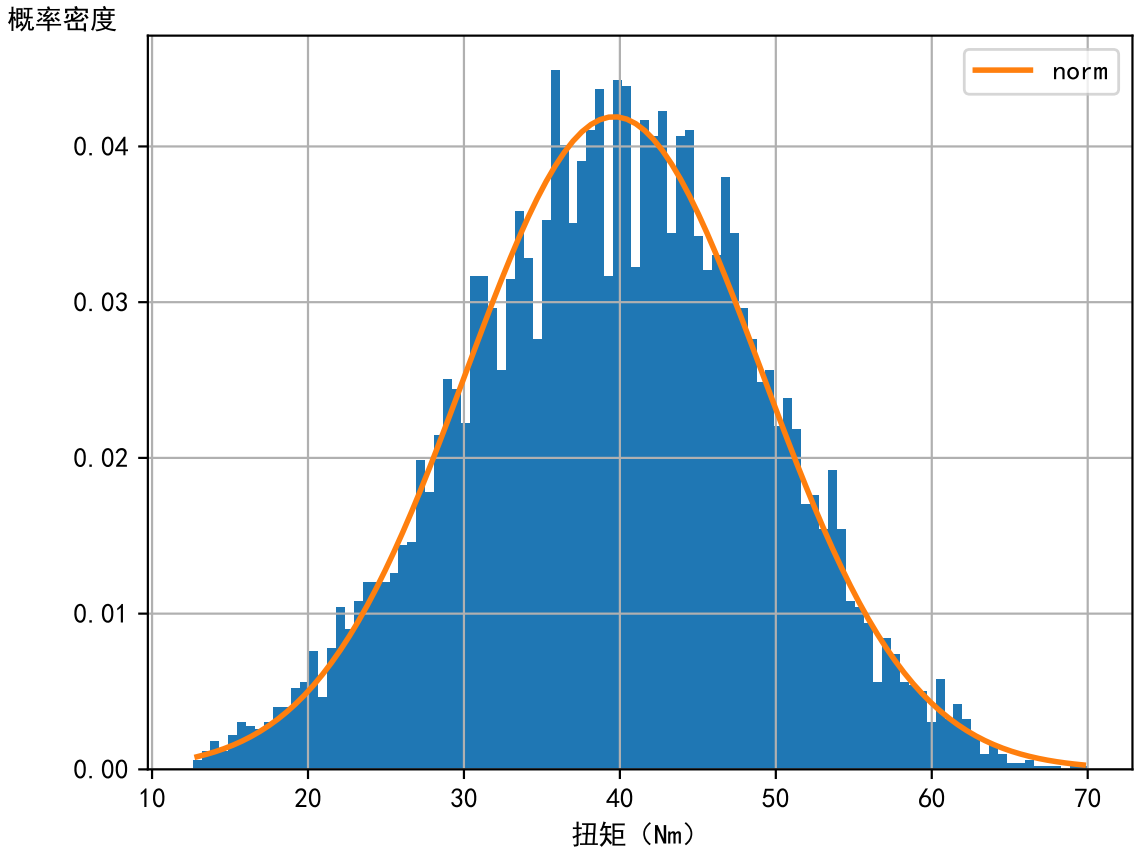

图 7: 扭矩的近似分布直方图

其中各个参数取值分别为 $loc = 39.623$，$scale = 9.520$。值得注意的是，对于正态分布，loc 与 scale 即为该正态分布的均值与标准差。

2.3.5　使用时长分布的拟合

经编程计算，得到拟合效果较好的分布如下：

分布	SSE	K-S 统计量	p 值
anglit	2.27×10^{-4}	0.06387	2.765×10^{-31}
cosine	2.79×10^{-4}	0.06128	7.786×10^{-29}
uniform	2.91×10^{-4}	0.1227	1.632×10^{-114}
rayleigh	2.99×10^{-4}	0.07024	9.484×10^{-38}
maxwell	3.11×10^{-4}	0.06646	8.101×10^{-34}

表 11: 使用时长分布的拟合结果

其中，uniform 代表均匀分布，rayleigh 代表瑞利分布，maxwell 代表 Maxwell 分布。我们选取**均匀分布**，作为非故障类机器使用时长的近似分布。

均匀分布的拟合效果如下图所示：

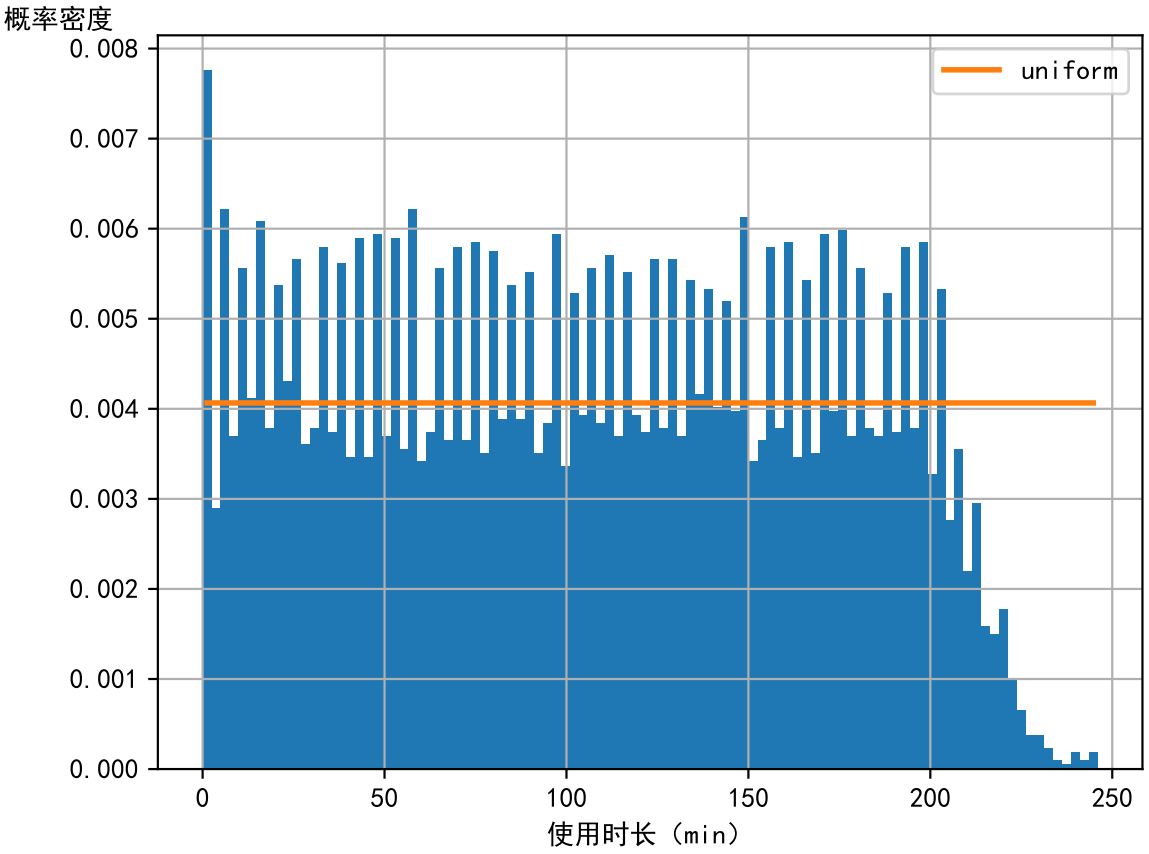

图 8: 使用时长的近似分布直方图

其中各个参数取值分别为 $loc = 0$，$scale = 246$。值得注意的是，对于均匀分布，loc

为该均匀分布的区间起点，scale 为该区间的长度。

从直方图中可以看出，使用均匀分布对使用时长进行拟合，K-S 检验不够显著的主要原因是，在某些使用时长的取值上，集中了较多的样本。这可能与该工厂生产安排有一定的关系。对于这类连续性较差的概率密度，我们使用均匀分布作为大致的近似，是可以接受的。

第 3 章　设备故障成因探究

3.1　基于列联分析检验的故障的成因探究

在第 2 章中，我们对五个连续型参数进行了初步的统计分析。对于定序的机器质量等级参数，我们无法直接套用连续型参数的分析方法。为探究各个故障的出现是否与机器质量等级存在关联，在本节中，我们将使用**列联分析检验**，依次检测每个故障类与质量等级的关联。

首先，我们对不同故障类别的机器质量等级绘制饼图，观测其大致的分布比例：

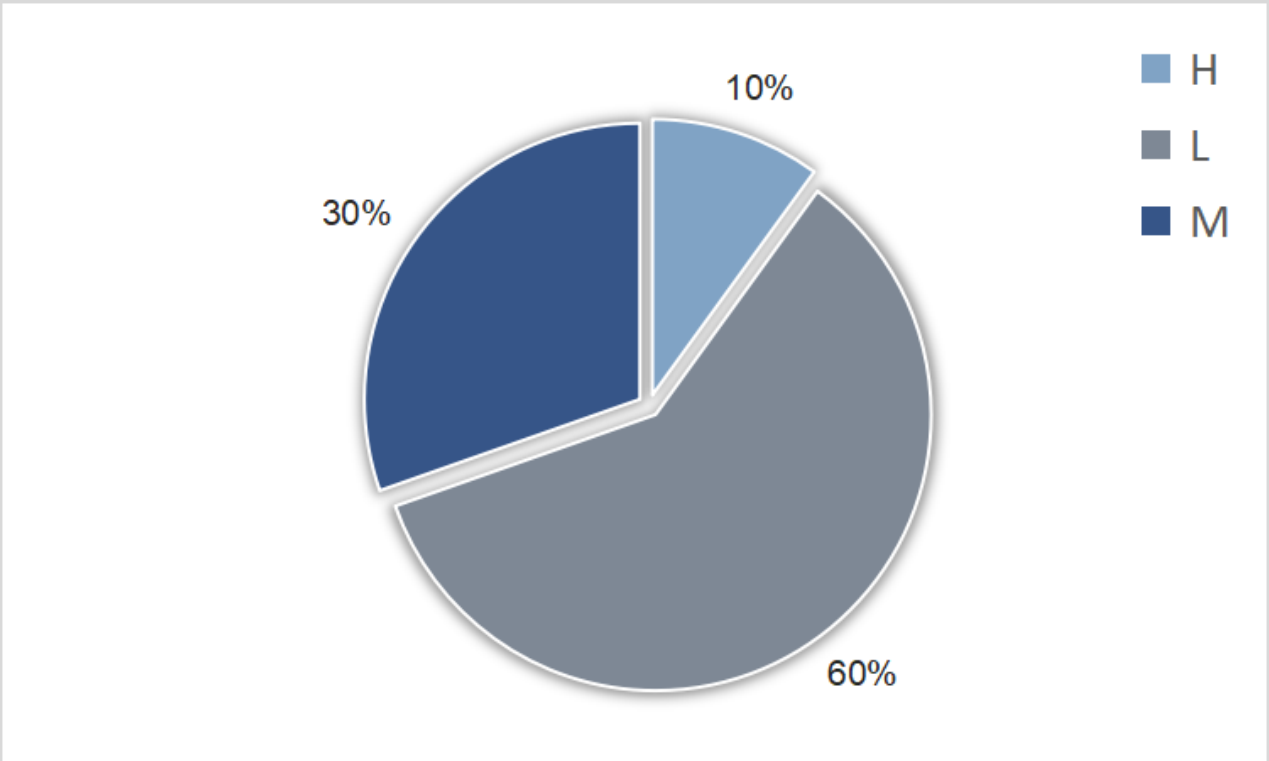

图 9: 非故障类机器质量等级的分布

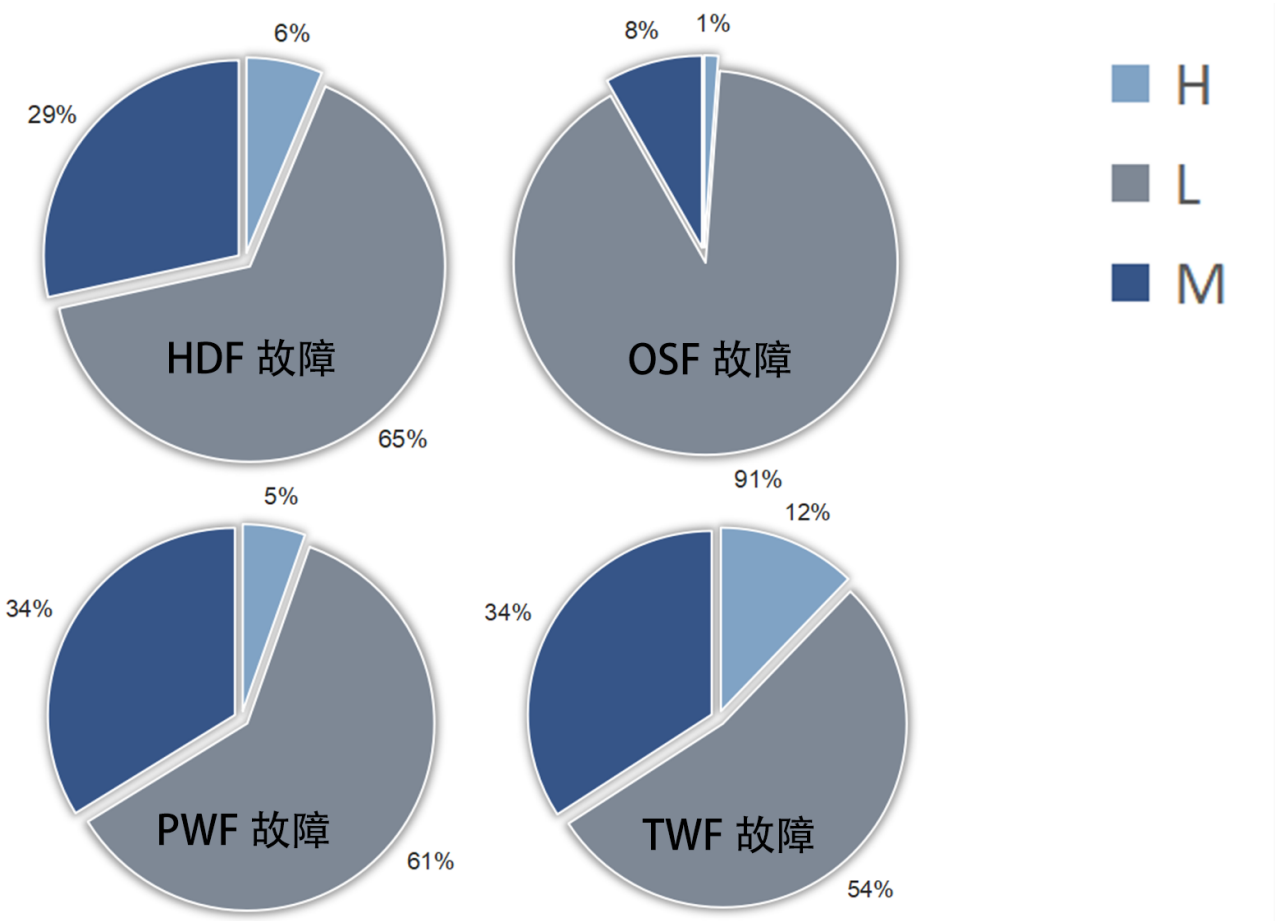

图 10: 各个故障类机器质量等级的分布

显然，OSF 类故障的质量等级分布与非故障类分布存在明显差异，而其他故障类别的机器质量等级分布与非故障类大致相同。

下面，我们使用列联分析检验的方法，定量检验各个故障类是否与机器质量等级有关。首先我们引入二维列联表的概念。二维列联表是指对观测个体的两个特征进行分类计算得到的频数分布表。设 n 个样本可按照两个特征 A（r 个水平）和 B（c 个水平）进行分类，二维列联表为：

	B_1	B_2	$\cdots$	B_c	合计
A_1	n_{11}	n_{12}	$\cdots$	n_{1c}	n_1
A_2	n_{21}	n_{22}	$\cdots$	n_{2c}	n_2
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
A_r	n_{r1}	n_{r2}	$\cdots$	n_{rc}	n_r
合计	$n_{\cdot 1}$	$n_{\cdot 2}$	$\cdots$	$n_{\cdot c}$	n

其中，n_{ij} 表示同时具有 A_i 和 B_j 特征的实际频数。用 p_{ij} 表示第 i 行、第 j 列格子的

理论频率，相应第 i 行和第 j 列的理论频率为 $p_{i\cdot} = \sum_{j=1}^{c} p_{ij}$ 和 $p_{\cdot j} = \sum_{i=1}^{r} p_{ij}$，且 n_{ij}/n 是 p_{ij} 的极大似然估计。当行变量和列变量独立时，有 $p_{ij} = p_{i\cdot}p_{\cdot j}$. 因此，分类变量间的独立性检验就可以转换为 $p_{ij} = p_{i\cdot}p_{\cdot j}$ 的拟合优度检验。在原假设成立时，应该有 $n_{ij} \approx \frac{n_i \cdot n_{\cdot j}}{n}$，其统计量称为 Pearson χ^2 统计量，即：

$$Q = \sum_{i=1}^{r}\sum_{j=1}^{c} \frac{(n_{ij} - n_{i\cdot} \cdot n_{\cdot j}/n)^2}{n_{i\cdot}n_{\cdot j}/n} \tag{17}$$

且在原假设下，可以证明 $Q \xrightarrow{L} \chi^2((r-1)(c-1))$

下面，我们提取每个故障类别，对不同的故障类别分别进行列联分析。以 OSF 类故障为例，我们令 A_1 取值为 L 级，A_2 取值为 M 级，A_3 取值为 H 级；B_1 取值为 OSF 类，B_1 取值为非 OSF 类。这样的列联分析检验，可以判断机器发生 OSF 类故障是否与机器质量等级有关。检验的结果如下表所示：

	HDF 故障	OSF 故障	PWF 故障	TWF 故障
p 值	0.4149	4.9319×10^{-8}	0.4022	0.6976

表 12: 各故障类与机器质量等级的列联相关检验

可以看到，除 OSF 类故障的列联分析检验显著拒绝了原假设，其余故障类别均可接受原独立性假设，这与饼图的初步分析结果一致。也就是说，**当机器质量较差时，机器出现 OSF 过载故障的概率将显著提高**，而出现其他故障的概率并没有明显改变。

3.2　基于 K-S 检验的故障的成因探究

在 2.3 中，我们已经介绍过 K-S 检验的基本原理。下面，我们利用 2.3 中已经拟合出的非故障类各个参数的近似分布函数，对故障类的各个参数进行 K-S 检验。**若一种故障的某个参数不符合相应的非故障类的近似分布，说明该参数的异常很有可能是导致该故障的原因**。根据这一原理，我们依次对四类故障进行分析。

3.2.1　TWF 类故障的成因探究

K-S 检验的 p 值如下：

	室温（K）	机器温度（K）	转速（rpm）	扭矩（Nm）	使用时长（min）
p 值	0.3315	0.8320	0.8051	0.5805	1.123×10^{-45}

表 13: TWF 类故障各个参数的 K-S 检验结果

针对显著拒绝原假设的参数，我们绘制其直方图如下：

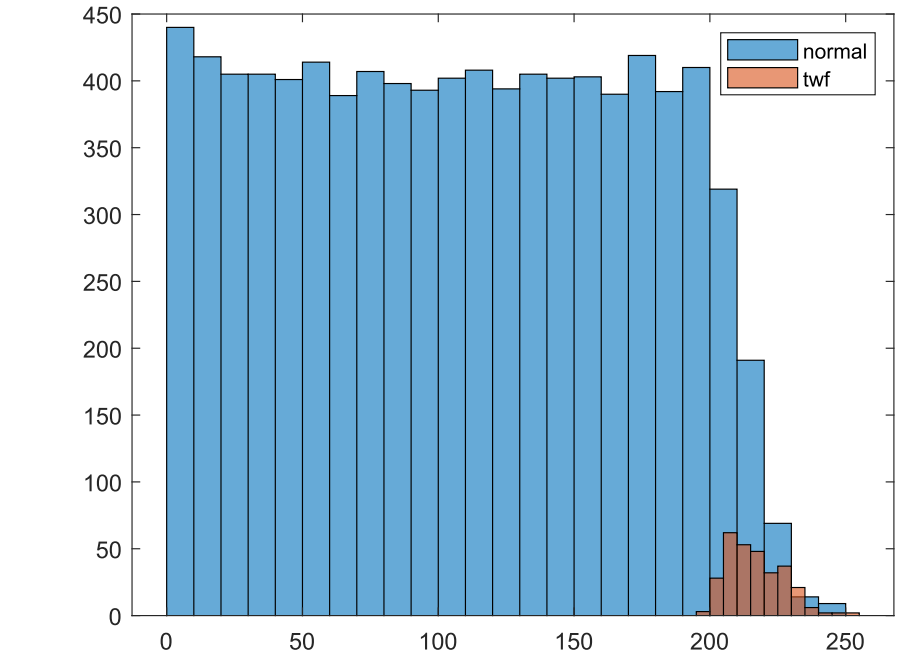

图 11: TWF 故障的使用时长直方图

通过直方图可以看出，发生 TWF 故障的设备，其平均使用时间显著高于非故障的设备。因此可以认为，**随着使用时间的增加，机械设备更容易发生 TWF 类磨损故障**。

3.2.2　HDF 类故障的成因探究

K-S 检验的 p 值如下：

	室温（K）	机器温度（K）	转速（rpm）	扭矩（Nm）	使用时长（min）
p 值	1.145×10^{-6}	4.552×10^{-14}	7.772×10^{-16}	7.772×10^{-16}	0.8026

表 14: HDF 类故障各个参数的 K-S 检验结果

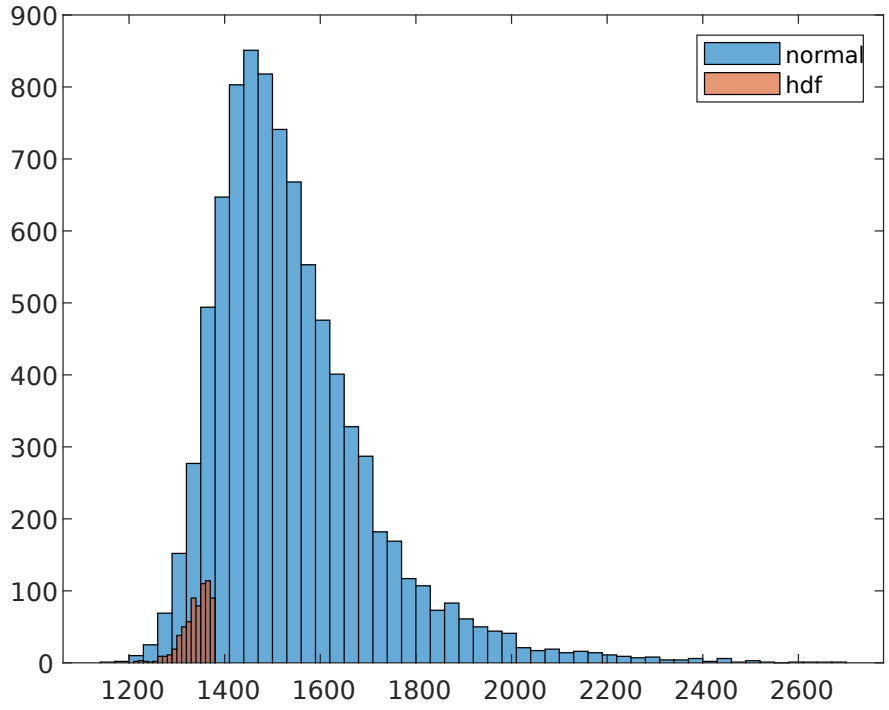

图 12: HDF 故障的转速直方图

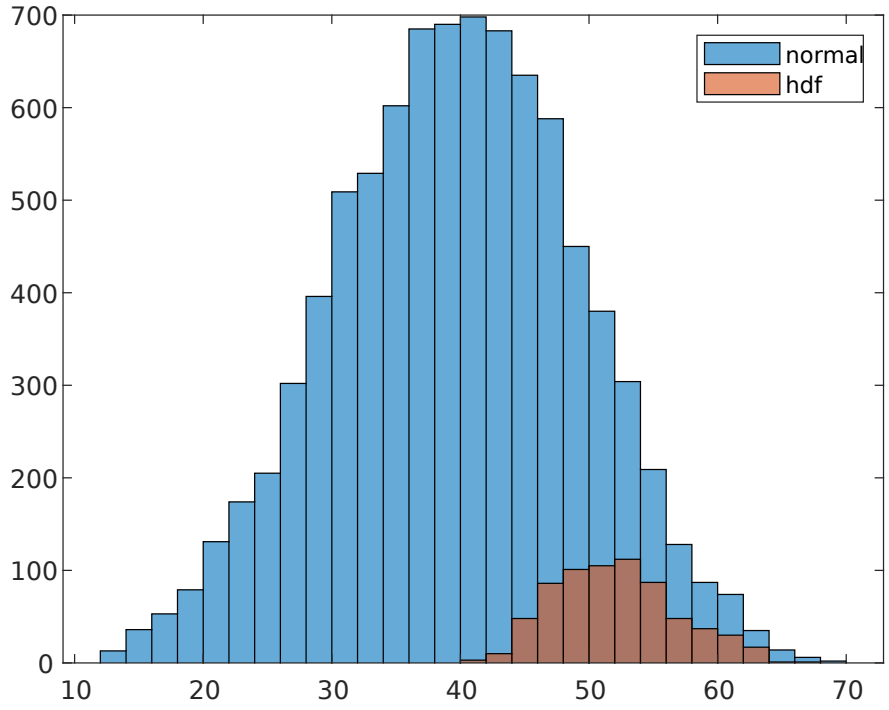

图 13: HDF 故障的扭矩直方图

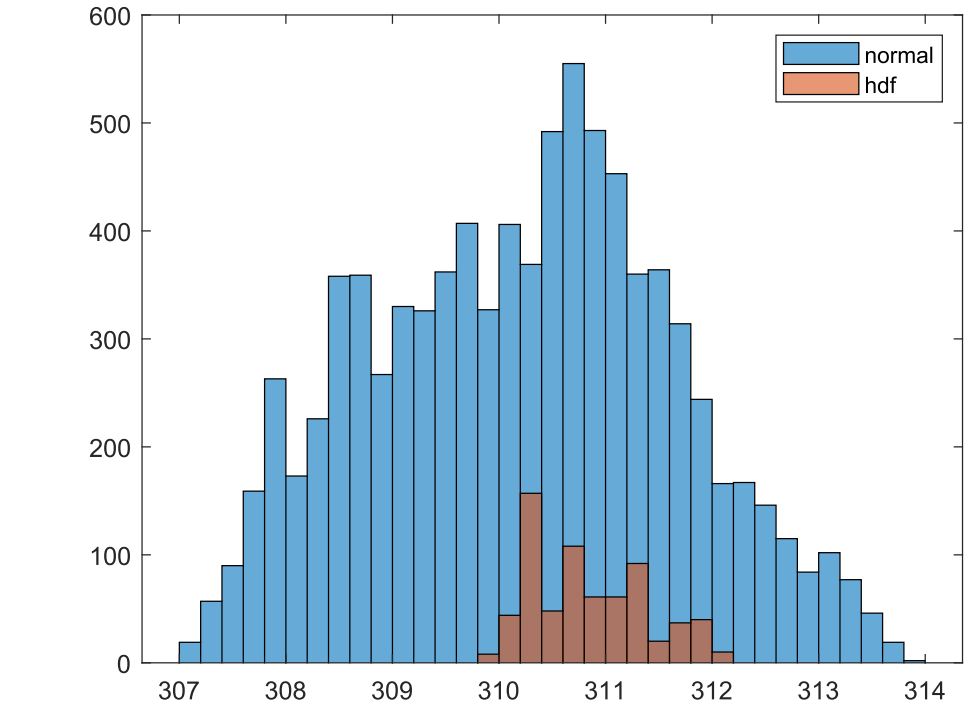

图 14: HDF 故障的机器温度直方图

针对显著拒绝原假设的参数，我们绘制其直方图如下：

从故障名称即可推测，HDF 类散热故障与温度有关。通过直方图进一步观测，可以看出，HDF 类故障设备的室温、扭矩与转速均明显区别于非故障类设备。也就是说，**对于机器温度集中在 $[310, 312]$ 范围内，转速较低扭矩较大的设备，更容易发生 HDF 类散热故障**。

3.2.3　PWF 类故障的成因探究

K-S 检验的 p 值如下：

	室温（K）	机器温度（K）	转速（rpm）	扭矩（Nm）	使用时长（min）
p 值	0.4215	0.5422	1.030×10^{-10}	1.665×10^{-15}	0.0400

表 15: PWF 类故障各个参数的 K-S 检验结果

针对显著拒绝原假设的参数，我们绘制其直方图如下：

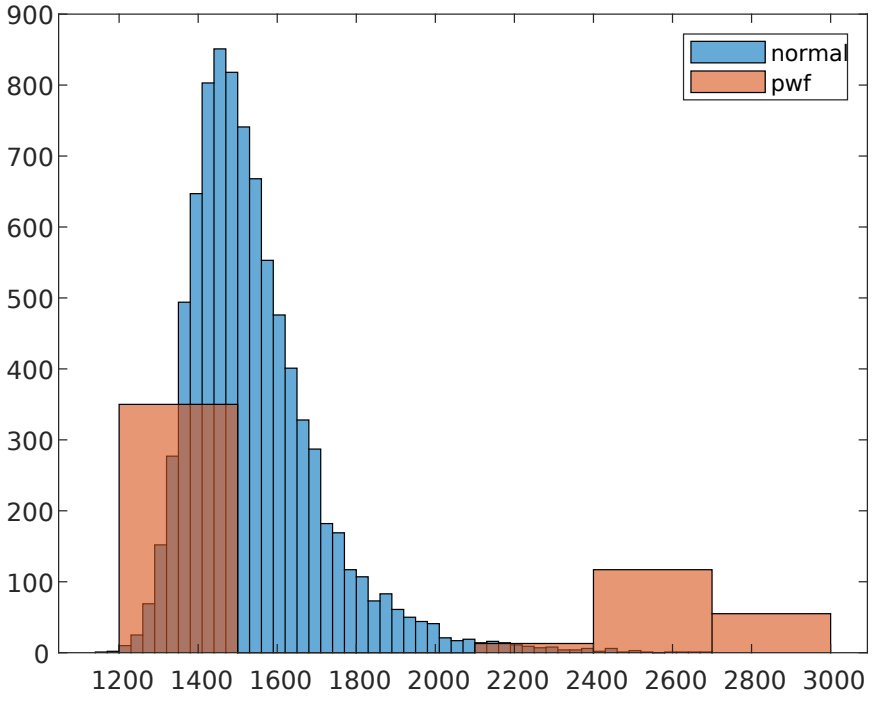

图 15: PWF 故障的转速直方图

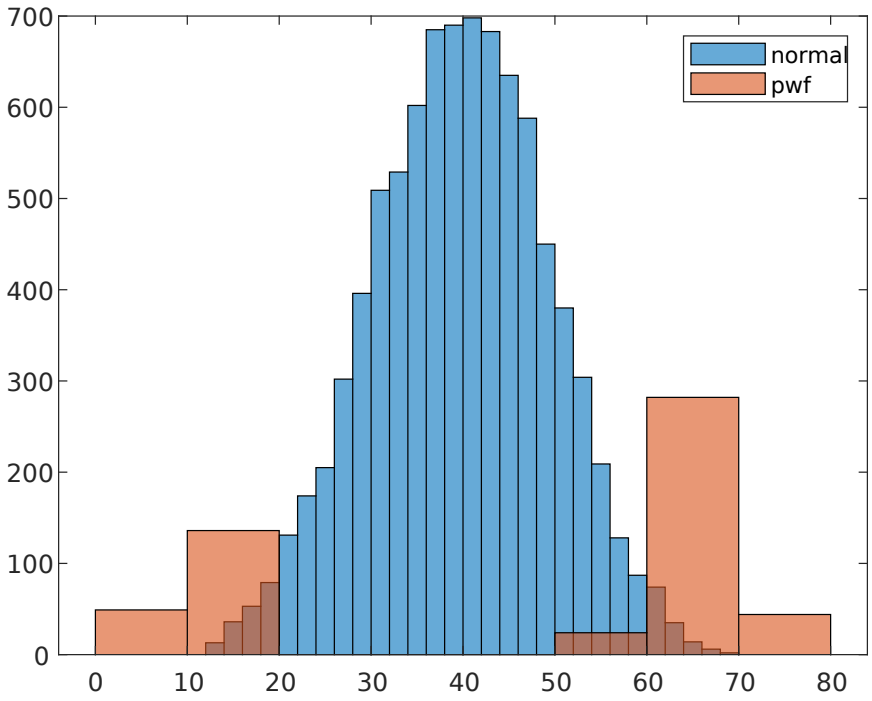

图 16: PWF 故障的扭矩直方图

出现 PWF 类故障的机械设备，其扭矩与转速的分布呈现两级分化的趋势。在直方图中可以清楚看到，PWF 类故障设备的扭矩明显高于或低于非故障类的平均水平。因此，**当机械设备的转速或扭矩明显偏低或偏高时，很有可能会发生 PWF 类电力故障**。

3.2.4　OSF 类故障的成因探究

K-S 检验的 p 值如下：

	室温（K）	机器温度（K）	转速（rpm）	扭矩（Nm）	使用时长（min）
p 值	0.9196	0.5544	2.109×10^{-15}	2.109×10^{-15}	1.919×10^{-10}

表 16: OSF 类故障各个参数的 K-S 检验结果

针对显著拒绝原假设的参数，我们绘制其直方图如下：

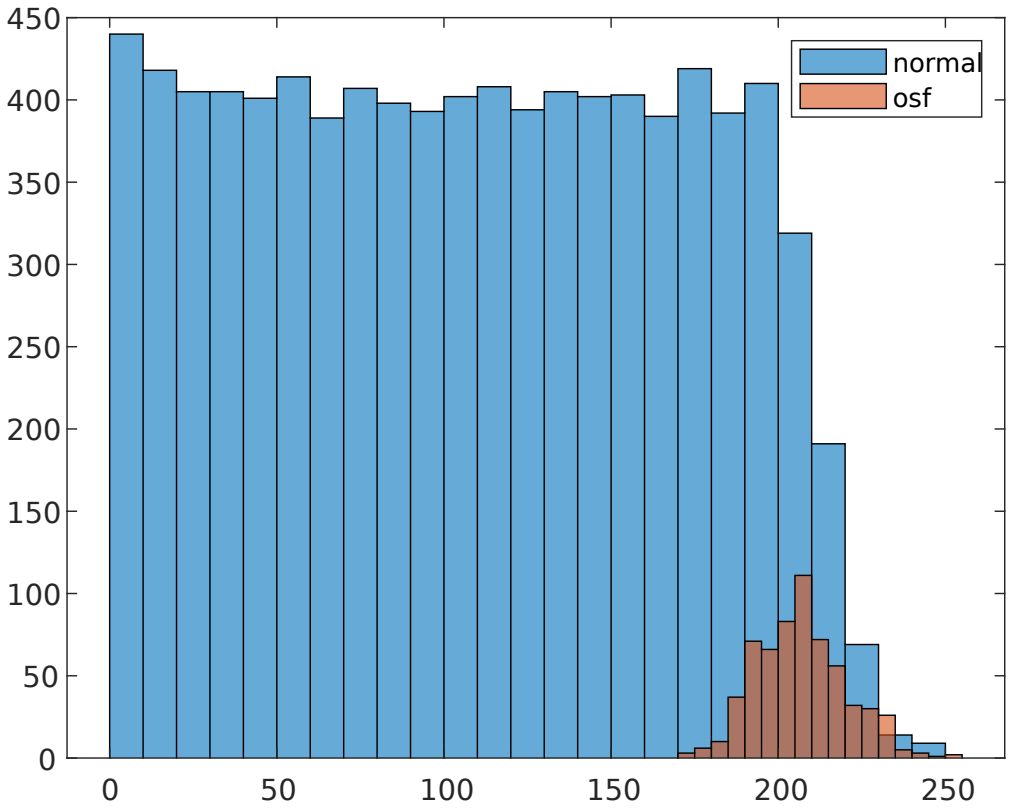

图 17: OSF 故障的使用时长直方图

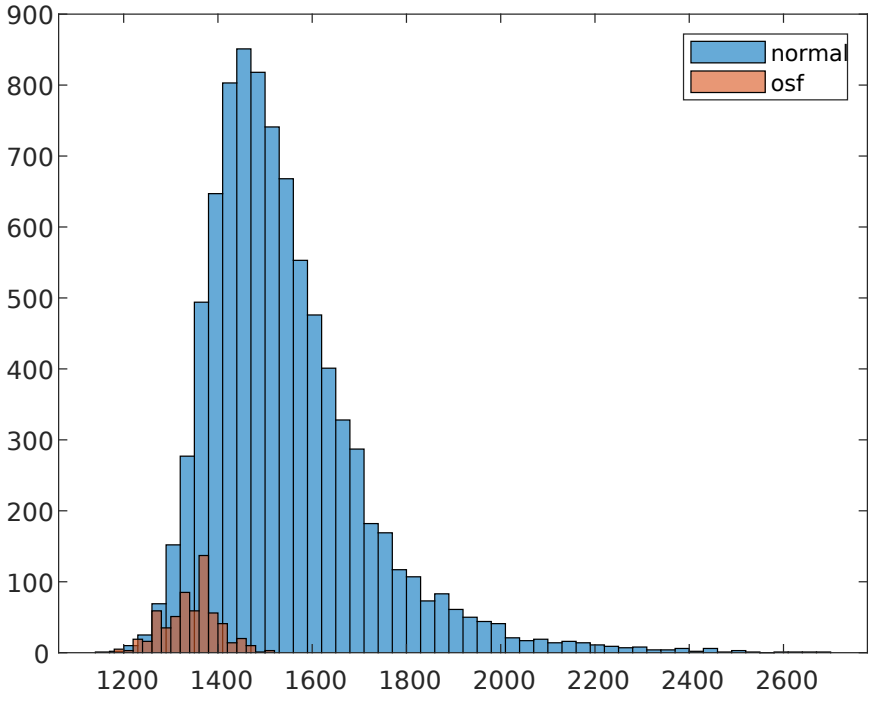

图 18: OSF 故障的转速直方图

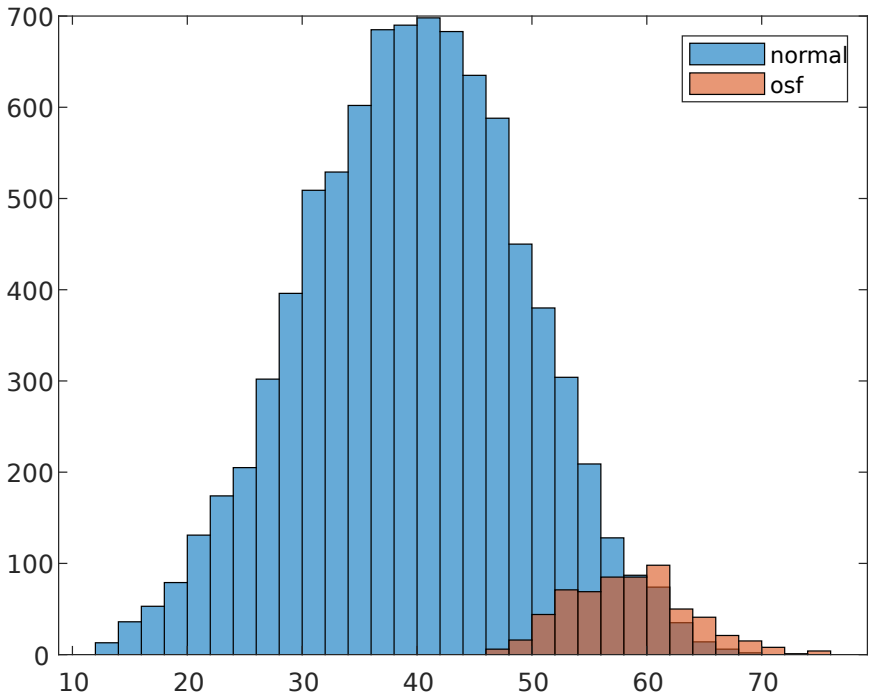

图 19: OSF 故障的扭矩直方图

通过直方图可以看出，发生 OSF 故障的机械设备其扭矩、转速与使用时长的分布，均不同于非故障类。值得注意的是，其他类的故障对于设备质量等级的分布是均匀的，而发生 OSF 类故障的设备，L 级占比很高。由此可以推测，**低质量等级的机械设备，若其他参数同时出现了异常，很有可能发生 OSF 类过载故障**。

第 4 章　结论

本项目使用某企业机械设备的使用情况及故障发生情况的数据集，基于参数估计、假设检验、Bootstrap 重抽样等统计计算方法，对机械设备的故障情况进行统计分析，最终给出了机械不同故障的可能成因。

在对非故障类设备分析后，我们得出以下结论：

- 机器温度与室温呈正相关性，转速与扭矩呈负相关性
- 室温与机器温度近似服从 Beta 分布、转速近似服从右拖尾耿贝尔分布、扭矩近似服从正态分布、使用时长近似服从均匀分布。

在对故障类设备分析后，我们得出以下结论：

- 随着使用时间的增加，机械设备更容易发生 TWF 类磨损故障
- 对于机器温度集中在 [310, 312] 范围内，转速较低扭矩较大的设备，更容易发生 HDF 类散热故障。
- 当机械设备的转速或扭矩明显偏低或偏高时，很有可能会发生 PWF 类电力故障。
- 低质量等级的机械设备，若其他参数同时出现了异常，很有可能发生 OSF 类过载故障。

参考文献

[1] 许王莉, 朱利平. 数据科学统计计算 [M]. 中国人民大学出版社，2022.

[2] 郑凡帆. 基于深度学习的机械设备故障预警与诊断技术研究 [D]. 北京化工大学,2020.DOI:10.26939/d.cnki.gbhgu.2020.000682.

[3] 百度百科. 偏度, https://baike.baidu.com/item/%E5%81%8F%E5%BA%A6/8626571?fr=aladdin [EB/OL].2020-04-01.

[4] 百度百科. 峰度, https://baike.baidu.com/item/%E5%B3%B0%E5%BA%A6/10840865?fr=aladdin [EB/OL].2021-01-25.

附录

2.1 相关系数检验的 SPSS 代码

```
CORRELATIONS
/VARIABLES=室温（K） 机器温度（K） 转速（rpm） 扭矩（Nm） 使用时长（min）
/PRINT=TWOTAIL NOSIG
/MISSING=PAIRWISE.

NONPAR CORR
/VARIABLES=室温（K） 机器温度（K） 转速（rpm） 扭矩（Nm） 使用时长（min）
/PRINT=BOTH TWOTAIL NOSIG
/MISSING=PAIRWISE.
```

```
GET
FILE='D:\统计计算\project\data.sav'.
DATASET NAME 数据集1 WINDOW=FRONT.
USE ALL.
COMPUTE filter_$=(具体故障类别 = "Normal").
VARIABLE LABELS filter_$ '具体故障类别 = "Normal" (FILTER)'.
VALUE LABELS filter_$ 0 'Not Selected' 1 'Selected'.
FORMATS filter_$ (f1.0).
FILTER BY filter_$.
EXECUTE.
CORRELATIONS
/VARIABLES=室温（K） 机器温度（K） 转速（rpm） 扭矩（Nm） 使用时长（min）
/PRINT=TWOTAIL NOSIG
/MISSING=PAIRWISE.

NONPAR CORR
/VARIABLES=室温（K） 机器温度（K） 转速（rpm） 扭矩（Nm） 使用时长（min）
/PRINT=BOTH TWOTAIL NOSIG
/MISSING=PAIRWISE.
```

2.2 矩统计量的数值估计的 Python 代码

```
import pandas as pd
import numpy as np



data = pd.read_excel('train_data.xlsx')
data = data.loc[data["是否发生故障"] == 0]
aa = data["机器质量等级"]
data = data.drop(columns=['机器编号', '统一规范代码', '机器质量等级',
    '是否发生故障', '具体故障类别'])
means = data.mean()
vars = data.var()
skews = data.skew()
kurts = data.kurt()
print('\n样本均值: \n')
print(means)
print('\n样本方差: \n')
print(vars)
print('\n样本偏度: \n')
print(skews)
print('\n样本峰度: \n')
print(kurts)

means_list = []
vars_list = []
skews_list = []
kurts_list = []
for i in range(1000):
x = data.sample(n=8992, replace=True)
means_list.append(x.mean())
vars_list.append(x.var())
skews_list.append(x.skew())
kurts_list.append(x.kurt())
```

```
32          means_list = np.array(means_list)
33          vars_list = np.array(vars_list)
34          skews_list = np.array(skews_list)
35          kurts_list = np.array(kurts_list)
36
37          predict_means = np.mean(means_list, axis=0)
38          predict_vars = np.mean(vars_list, axis=0)
39          predict_skews = np.mean(skews_list, axis=0)
40          predict_kurts = np.mean(kurts_list, axis=0)
41          bias_means = predict_means - means
42          bias_vars = predict_vars - vars
43          bias_skews = predict_skews - skews
44          bias_kurts = predict_kurts - kurts
45          se_means = np.std(means_list - predict_means, axis=0) + means -
                means
46          se_vars = np.std(vars_list - predict_vars, axis=0) + vars - vars
47          se_skews = np.std(skews_list - predict_skews, axis=0) + skews -
                skews
48          se_kurts = np.std(kurts_list - predict_kurts, axis=0) + kurts -
                kurts
49
50
51          print('\nBootstrap估计样本均值的偏差: \n')
52          print(bias_means)
53          print('\nBootstrap估计样本方差的偏差: \n')
54          print(bias_vars)
55          print('\nBootstrap估计样本偏度的偏差: \n')
56          print(bias_skews)
57          print('\nBootstrap估计样本峰度的偏差: \n')
58          print(bias_kurts)
59          print('\nBootstrap估计样本均值的标准差: \n')
60          print(se_means)
61          print('\nBootstrap估计样本方差的标准差: \n')
62          print(se_vars)
```

```
print('\nBootstrap估计样本偏度的标准差: \n')
print(se_skews)
print('\nBootstrap估计样本峰度的标准差: \n')
print(se_kurts)

print('\nBootstrap估计样本均值与方差的置信区间: ')
columns = data.columns
for i in range(5):
left_means = 2 * means[i] - np.percentile(means_list[:, i], 97.5)
right_means = 2 * means[i] - np.percentile(means_list[:, i], 2.5)
left_vars = 2 * vars[i] - np.percentile(vars_list[:, i], 97.5)
right_vars = 2 * vars[i] - np.percentile(vars_list[:, i], 2.5)
left_skews = 2 * skews[i] - np.percentile(skews_list[:, i], 97.5)
right_skews = 2 * skews[i] - np.percentile(skews_list[:, i], 2.5)
left_kurts = 2 * kurts[i] - np.percentile(kurts_list[:, i], 97.5)
right_kurts = 2 * kurts[i] - np.percentile(kurts_list[:, i], 2.5)
print('\n' + str(columns[i]) + ' 置信水平为95%的置信区间: ')
print('均值: [' + str(left_means) + ',' + str(right_means) + ']')
print('方差: [' + str(left_vars) + ',' + str(right_vars) + ']')
print('偏度: [' + str(left_skews) + ',' + str(right_skews) + ']')
print('峰度: [' + str(left_kurts) + ',' + str(right_kurts) + ']')
```

2.3 非故障类特征的拟合分布的 Python 代码

```
import pandas as pd
import matplotlib.pyplot as plt
from fitter import Fitter

data = pd.read_excel("train_data.xlsx", engine='openpyxl')
data = data.drop(["机器编号"], axis=1)
data = data.drop(["是否发生故障"], axis=1)
data_normal = data.loc[data["具体故障类别"] == 'Normal']

```

```
10          def print_hist(f):
11          # 返回最佳拟合分布及其参数
12          print(f.get_best(method='sumsquare_error'))
13          # 绘制组数=bins的标准化直方图
14          plt.figure()
15          f.hist()
16          f.plot_pdf(lw=2, method='sumsquare_error')
17
18          plt.rcParams['font.sans-serif'] = ['SimHei'] # 用来正常显示中文标签
19          plt.rcParams['axes.unicode_minus'] = False # 用来正常显示负号
20
21          f_temperature1 = Fitter(data_normal["室温（K）"].values, timeout=60,
                distributions=['beta'])
22          f_temperature1.fit()
23          print(f_temperature1.summary(plot=False))
24          print_hist(f_temperature1)
25          plt.xlabel("室温（K）")
26          plt.ylabel("概率密度", rotation='horizontal',
                verticalalignment='top', horizontalalignment='center', y=1.04)
27          plt.show()
28
29          f_temperature2 = Fitter(data_normal["机器温度（K）"].values,
                timeout=60, distributions=['beta'])
30          f_temperature2.fit()
31          print(f_temperature2.summary(plot=False))
32          print_hist(f_temperature2)
33          plt.xlabel("机器温度（K）")
34          plt.ylabel("概率密度", rotation='horizontal',
                verticalalignment='top', horizontalalignment='center', y=1.04)
35          plt.show()
36
37          f_rotate = Fitter(data_normal["转速（rpm）"].values, timeout=60,
                distributions=['gumbel_r'])
38          f_rotate.fit()
```

```
print(f_rotate.summary(plot=False))
print_hist(f_rotate)
plt.xlabel("转速（rpm）")
plt.ylabel("概率密度", rotation='horizontal',
    verticalalignment='top', horizontalalignment='center', y=1.04)
plt.show()

f_torque = Fitter(data_normal["扭矩（Nm）"].values, timeout=60,
    distributions=['norm'])
f_torque.fit()
print(f_torque.summary(plot=False))
print_hist(f_torque)
plt.xlabel("扭矩（Nm）")
plt.ylabel("概率密度", rotation='horizontal',
    verticalalignment='top', horizontalalignment='center', y=1.04)
plt.show()

f_time = Fitter(data_normal["使用时长（min）"].values, timeout=60,
    distributions=['uniform'])
f_time.fit()
print(f_time.summary(plot=False))
print_hist(f_time)
plt.xlabel("使用时长（min）")
plt.ylabel("概率密度", rotation='horizontal',
    verticalalignment='top', horizontalalignment='center', y=1.04)
plt.show()
```

3.1 列联分析检验的 Python 代码

```
import pandas as pd
import numpy as np
from scipy.stats import chi2_contingency

```

```
data = pd.read_excel('train_data.xlsx')
# data = data[['机器质量等级','具体故障类别']]
matrix = data.groupby(['机器质量等级','具体故障类别']).count()
print(matrix)
matrix = np.array(matrix)
# print(matrix)
matrix = matrix[:,0].reshape(3,5)
labels = ['HDF','Normal','OSF','PWF','TWF']
print(matrix)

for i in range(len(matrix[0])):
newmatrix = np.vstack((matrix.sum(axis=1) - matrix[:, i], matrix[:,
    i]))
newmatrix = np.transpose(newmatrix)
chi, p, v, exp = chi2_contingency(newmatrix)
print('''
==========================================================
matrix:
{}
报告
{}故障类型与非{}故障类型
 2值:{}
p值:{}
自由度:{}
理论值:
{}
==========================================================
'''.format(newmatrix,labels[i],labels[i],chi, p, v, exp))
```

3.2 K-S 检验 Python 代码

```
from scipy.stats import norm, chi2, mstats, shapiro, kstest,
    chisquare
```

```
import numpy as np
import pandas as pd
from sympy import *
from scipy.stats import beta,gumbel_r,uniform

def pearson_chi_squared(col1, col2, alpha):

print('\n Defining Hypothesis')
print('\n Null Hypothesis: There is no association between',
    col1.name, 'and', col2.name)
print('\n Alternative Hypothesis: There is an association between',
    col1.name, 'and', col2.name)

table = pd.crosstab(col1, col2, margins=False)

sub_component = 0
e_frequencies = []
freq_less_than_five = []
freq_less_than_or_equal_one = []

for i in range(0, len(table.index)):
for j in range(0, len(table.columns)):
expected_frequency = (table.iloc[:, j:(j + 1)].sum().sum() *
    table.iloc[i: i + 1, :].sum(axis=1).sum()) \
/ table.sum(axis=1).sum()
e_frequencies.append(expected_frequency)
observed_frequency = table.iloc[i, j]
sub_component = ((observed_frequency - expected_frequency) ** 2 /
    expected_frequency) + sub_component
if expected_frequency < 5:
freq_less_than_five.append((i, j))

elif expected_frequency <= 1:
freq_less_than_or_equal_one.append((i, j))
```

```
32          print(len(e_frequencies))
33          e_ratio = len(freq_less_than_five) / len(e_frequencies)
34          e_len = len(freq_less_than_or_equal_one)
35
36          test_Statistics = sub_component
37          degrees_of_freedom = [(len(table.columns) - 1) * (len(table.index)
                - 1)]
38          a = alpha / 100
39          chi_critical = chi2.isf(q=a, df=degrees_of_freedom)
40          p_value = chi2.sf(test_Statistics, degrees_of_freedom)
41
42          print('\n Rejection Criteria: Reject Null Hypothesis if Test
                Statistic is greater than or equal to Critical Value '
43          'at', alpha, '% level of Significance.')
44          print('\n Test Results')
45          print('\n', pd.crosstab(col1, col2, margins=True))
46          results = {'Categorical Variable 1': col1.name ,'Categorical
                Variable 2': col2.name, 'Test Statistic': round(test_Statistics,
                4),
47              'Critical Value': round(chi_critical[0], 4), 'P value':
                    p_value[0]}
48          print('\n', results)
49
50          if e_ratio <= 0.2:
51          print('\n Note : No more than 20% of expected frequencies are less
                than 5.')
52
53          elif e_ratio > 0.2:
54          print('\n Warning : More than 20% of expected frequencies are less
                than 5. Hence, Chi-Squared Critical '
55          'Value is invalidated by small expected frequencies.')
56
57          elif e_len >= 1:
58          print('\n Warning :', len(freq_less_than_or_equal_one),
```

```
59          'number of expected frequency with less than or equal one is
                indicated.'
60          'Hence, Chi-Squared Critical Value is invalidated by small expected
                frequency.')
61
62          elif e_ratio > 0.2 and e_len >= 1:
63          print('\n Warning : More than 20% of expected frequencies are less
                than 5 and '
64          'also ', len(freq_less_than_or_equal_one),
65          'number of expected frequency with less than or equal one is
                indicated. '
66          'Hence, Chi-Squared Critical Value is invalidated by small expected
                frequencies.')
67
68
69          """Applying Pearson's chi square test for Sepsis data"""
70
71          # print("\n Results for Sepsis data")
72          #
73          # data2 = pd.read_csv("Sepsis.csv")
74          # column1 = data2['THERAPY']
75          # column2 = data2['Age_Group']
76          #
77          # pearson_chi_squared(column1, column2,alpha = 5)
78          #
79          #
80          # """Applying Pearson's chi square test for product and payment
                data"""
81          #
82          # print("\n Results for product and payment data")
83          #
84          # data1 = pd.read_csv("Product and Payment.csv")
85          # column3 = data1['Type of Product']
86          # column4 = data1['Type of Payment']
```

```
87          #
88          # pearson_chi_squared(column3, column4,alpha = 5)
89
90          data = pd.read_excel('./train_data.xlsx')
91          normal_data = data[data['具体故障类别'] == 'Normal']
92          environ_temp = normal_data['室温（K）']
93          Abnormal_data = data[data['具体故障类别'] == 'HDF']
94          environ_temp1 = Abnormal_data['室温（K）']
95          mach_temp = Abnormal_data['机器温度（K）']
96          pearson_chi_squared(mach_temp,environ_temp1,alpha=5)
```

小组分工

200810205-林世铮：负责 *3.2* 基于 *K-S* 检验的故障的成因探究的 Python 编程实现；负责对假设检验的效果作分析。

200810301-李岳锴：寻找数据集并进行预处理；负责 *2.3* 非故障类特征的拟合分布分布的 Python 编程实现；整理其他成员的编程输出并进行图表可视化；负责全部报告的写作。

200810229-李毓晟：负责 *2.2* 矩统计量的数值估计的 Python 编程实现；负责全部 PPT 的制作。

200810225-林瑞奇：负责 *2.1* 相关系数检验的 SPSS 编程实现、*3.1* 基于列联分析检验的故障成因的探究的 Python 编程实现；绘制图 9、图 10 的饼图。