2. Forecasting

Reorder point = demand · Leadtime + service level · std · $\sqrt{\text{Lead time}}$

Casual & Time-series Method.

↳ system variable

↳ derive past/future

① moving average

回看过去 n 个 period.

n 越小，越 responsive

② simple exp smoothing

Forecast today = α · Demand yest + (1-α) Forecast yest

α ↑, responsive ↑

if 原数据有 trend，则预测会滞后

③ double exp smoothing

$F_{t+1} = S_t + T_t$

S_t (Forecast constant) = $\alpha Demand_t + (1-\alpha) F_t$

T_t (Forecast trend) = $\beta(F_t - F_{t-1}) + (1-\beta)T_{t-1}$

$F_{t+1} = \alpha Demand_t + (1-\alpha) F_t + \beta(F_t - F_{t-1}) + (1-\beta)T_{t-1}$

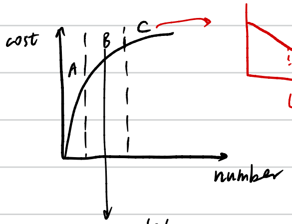
④ problems，预测出现极端值

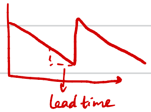
Sol: ABC analysis

different customer → different pattern

get advanced notice

3. Inventory





EOQ model

↳ minimize holding cost + fixed order cost

① fixed order/setup cost K

② holding cost h (Interest rate)

③ Total Demand D

⇓

Best Quantity per time

$Q = \sqrt{\frac{2KD}{h}}$

Total cost = $K \cdot \frac{D}{Q} + \frac{Q}{2} \cdot h$

↓ 产生规模收益 ⟶ 同时 safety stock 也会减少

2 factories ⇒ $Q = \sqrt{\frac{2KD}{h}} \cdot 2$

combine ⇒ $Q = \sqrt{\frac{2K \cdot (2D)}{h}}$

& save money

⇓

$z \cdot \sigma_1 \sqrt{L} + z \sigma_2 \sqrt{L}$

$> z \cdot \sqrt{\sigma_1^2 + \sigma_2^2} \cdot \sqrt{L}$

若 Demand 有 σ

ROP ⇒ $D \cdot L + z \cdot \sigma \cdot \sqrt{L}$

⇒ Average inventory = $z \cdot \sigma \cdot \sqrt{L} + \frac{Q}{2}$

(z: one-tail z 值，与 service level 有关)

若 leadtime 也有 σ

⇒ ROP = $D \cdot L + z \cdot \sqrt{D^2 \sigma_L^2 + \sigma_D^2 \cdot L}$

4. Newsvendor prob.

service level：累积概率（满足 Demand 要求）

To maximize profit: $c \cdot F(Q) = r \cdot (1 - F(Q))$

c 是 holding cost，r 是无满足的 shortage cost（少挣多少钱）revenue − cost

↳ $F(Q) = \frac{r}{c+r}$

+ salvage value（回收）

↳ $F(Q) = \frac{r-c+s}{r-c+s+c+h-v} = \frac{r-c+s}{r+s+h-v}$

c: cost
r: revenue
s: shortage cost
h: holding cost
v: salvage.

↳ $Q = mean + z \cdot \sigma$ (z = qnorm(F(Q)))

expected lost sales = $\sigma \cdot$ (P(service level) − z(service level) · (1 − service level))

↳ expected sales = mean − expected lost sales

expected leftover = Q − expected sales

↓

expected profit = (profit − cost) · sales − (cost − salvage) · leftover

5.

① Little's Law: $I = R \cdot T$

I: inventory, R = flow rate, T = flow time

Turn = $\frac{1}{T} = \frac{R}{I}$

② Bottle Neck capacity → max flow rate

Multi-stage with buffer

③ 专有名词:

Activity time / run time 完成项目的 time

Capacity: 1/Act Time Flow unit per time

↳ BottleNeck 最小 cap 地方

Cycle time: 完成 unit 所需时间

Flow rate: min{Demand rate, process capacity}

Flow time: 一单位所需时间

takt time: 可用时间 / Demand

utilization: flow rate / capacity

thruput rate: 1 / cycle time

④ 三种处理库存方式

Make to order

Assemble to order

Make to stock

↳ 5 个要素:

Process choice — continuous process-project

Vertical integration 参与议价多少

Resource flexibility

customer involvement

capital intensity 自动化

6. scheduling

①. ordered, partially ordered

②. sequencing rule

single processor
- ①. FCFS 先到先得 按顺序
- ⇒ ②. EDD earliest Due Date
- ③. Least changeover time
- ④. Critical Ratio : $\frac{\text{离Due date时间}}{\text{剩余生产时间}}$ 越小,优先级越高
- ⑤. SPT: shortest processing time 生产时间短的优先

③. 2-machine: Johnson's Rule

	A	B	C	D	E
A	①	②	3	4	5
B	5	4	③	②	①

A B C D E

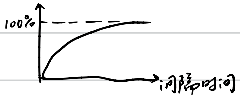
一组时间短的排前面
另一组时间短的排后面

④. Appointment:
平衡过度booking与过度闲置的成本

7. Queuing

①. Arrival Process (exp)

概率 $F(t)=1-e^{-\lambda t}$ λ是平均arrival rate



poisson : exp 镜像
λ是平均arrival rate

②. service process

$F(t)=1-e^{-\frac{t}{s}}$ → 平均service time

③. Variable
- λ: average arrival rate
- S: average service time
- σ^2: variance of t
- L: average arrival number to system
- Lq: average customer number in queue
- W: average time arrival in system
- Wq: average time arrival wait in queue

④. Little's law : $L=\lambda\cdot W$ 平均顾客数量

For queue, we assume $\lambda\cdot s<1$ 不然queue长度↑

$L_q=\frac{\lambda^2 s^2}{1-\lambda s}$ $W_q=\frac{L_q}{\lambda}=\frac{\lambda s^2}{1-\lambda s}$

$L=L_q+\lambda s=\frac{\lambda s}{1-\lambda s}$ $W=\frac{L}{\lambda}=\frac{s}{1-\lambda s}$

若service time有σ (不满足exponential)

$\Rightarrow L_q=\frac{\lambda^2(s^2+\sigma^2)}{2(1-\lambda s)}$

8. variability

utilization < 100% → stable

↳ Erlang C 算客需要等待的概率

①. 用priority rule (但设计时间可能不准)

②. pooling → reduce time in queue
因为separate可能同时wait & idle

↳问题
①需要 broader skills
②. 比其他人wait代价提升整体
③. 可能打断顾客与某-server的关系.

9. Quality Management

①. kaizen: 持续改善

②. Deming wheel: plan, do, check, act

③. six-sigma model:
$C_p=\frac{USL-LSL}{6\sigma}$ =2 时满足6σ standard.

④. Big batch lower quality
: 若有次品,则发现时损失较大

Inspect 第一次生产出时,在bottleneck前检测能提高效率

⑤. Shingo System
- ①. reduce setup time
- ②. Achieve 0 defect

⑥. Benchmarking : ISO 9000, 14000

10. Just in time production

① 降低inventory 能让问题暴露
↓
长sc, 长lead time, hide variability

Big setup ↑ variance ↑

JIT目标降低waste
- Transportation
- Inventory
- Motion
- Waiting
- Over-production
- over-processing
- Defect

Reduce setup 非常必要
收益很高

知道Q → 实现的成本

②. push → pull system
↑ produce asap ↑ downstream ask

③. JIT 直接发现错误 Detect, Alert, Stop

Jidoka: 自动化, 源头的品质

poya yoke: 防故障

Heijunka: 平顺化,降低variability
但多技 workforce,供应商有更高要求

kanban : 有看板→生产 ⟶ 数量 = $\frac{D\cdot L(1+S)}{C}$
- Average Demand ↑, lead time, safety stock, container size

↳ 2-card kb sys: C kanban 授权 take container
P kanban 授权生产

Lot size 可能大于 container

JIT关键
- ①. 小 setup cost
- ②. stable 生产 schedule
- ③. flexible, cross-train

11. project management

Critical path: 最长的路径
类似Dijkstra算法,找到最长路径
↳ Mean最大,但一般我们汇报时要计算n天内完成的prob

PERT 考虑 Mean 和 var ⟶ 但也有问题
非常难情况较难考虑

$Mean=\frac{a+4m+b}{6}$ (a 最乐观, m 最可能, b 最悲观)

$Var=\left(\frac{b-a}{6}\right)^2$ $std=\frac{b-a}{6}$

12. SC variability

Bullwhip Effect: SC var 会逐层放大

EOQ, 计划偏高, Forecast方法, price change
都会增大sc var

若有信息交互则能有global optim

减小Bullwhip Effect
- ①. smooth product flow
- ②. Eliminate bad incentives 用原因避免加班
- ③. Vendor-Managed Inventory

合作减少sc var
- ①. share data, 管理库存
- ②. 集中规划
- ③. Use everyday
- ④. Make small orders
- ⑤. Lower lead time

Buy-back contract
回购未卖完货物

Options contract
签期权合同

Contract with 不对等信息
买方良供应商共同规划,
但买方有夸大嫌疑
↳
- ①. Capacity reservation合同
买方支付供应商容量预留费用
- ②. Advanced purchase contract
买方提前支付订单预付金额