

Table of Contents

01



데이터 설명

02



단변량 분석

03



이변량 분석

04

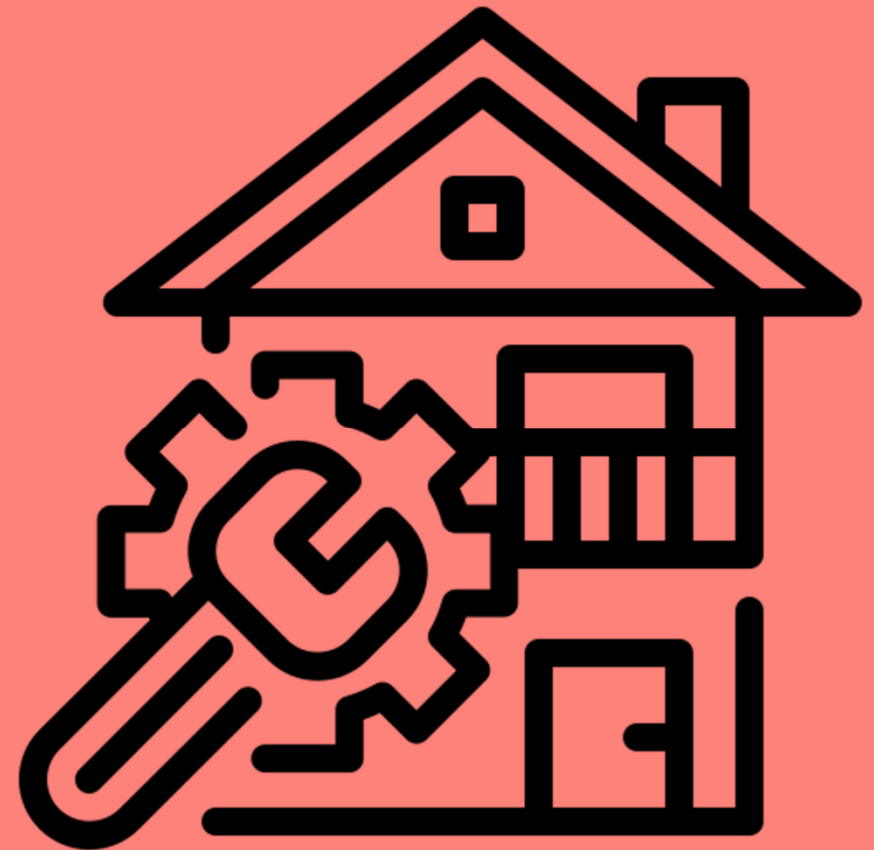


프로세스 마이닝

05



프로세스 개선 방안



데이터 설명

데이터 설명

변수명	내용	구분
CaseID	집 수리 요청 접수 번호	int
TaskID	업무	Object
Originator	업무 담당자	Object
Eventtype	업무의 시작(start)과 끝(complete)을 표시	Object
Contact	집 수리 요청의 요청 채널('Phone','Web','Personal','Letter')	Object
RepairType	수리 방식('P','E','B')	Object
ObjectKey	수리 대상 집 key	float
RepairInternally	내부 수리 여부('True','False')	object
EstimatedRepairTime	예상 수리 시간(60.0 ~ 480.0)	float
RepairCode	수리 종류(1.0 ~ 4.0)	float
RepairOK	수리 정상 종료 여부('True')	object
Date	업무 수행 일자	object
Time	업무 수행 시각	object

- 빨강색 : 프로세스 마이닝을 위해 CaseID, TaskID, DateTime(Date + Time) 컬럼을 Event에 활용
- 주황색 : 병목의 직접적인 원인이 될 것 같아 단변량 + 이변량 분석
- 나머지 데이터는 결측치가 많고 의미가 없어서 생략

단변량 분석

단변량 분석

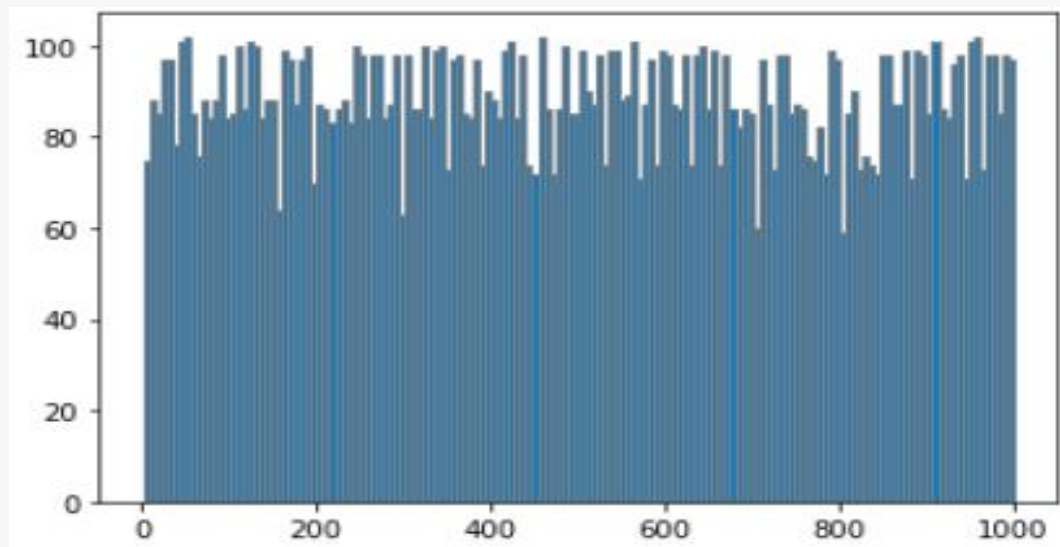


caseID

taskID

Originator

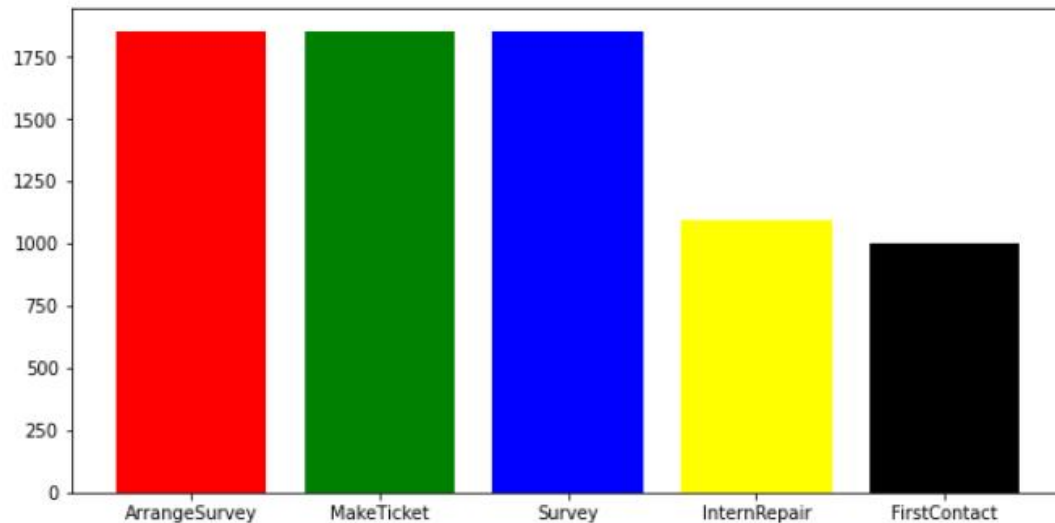
단변량 분석



caseID

1. caseID의 종류별 count를 그래프 표현
2. caseID의 종류는 1000개이며 기본적으로 여러 번 집수리 요청을 할 수 있음을 알 수 있음

단변량 분석

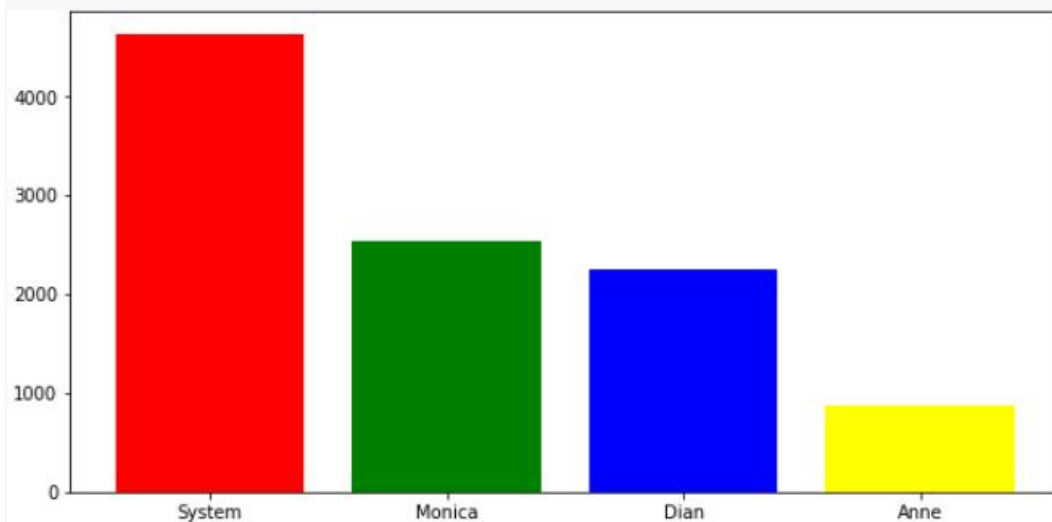


TaskID

1. TaskID의 종류별 count를 그래프 표현
2. 결측지 부분만 제거 하고 표현
3. 상위 5개만 표현

- 가장 많이 업무는 ArrangeSurvey, MakeTicket, Survey가 1854
- 가장 적은 업무는 ExternRepair, InformClientWrongPlace
- 만약 상위 업무의 직원이 적다면 병목현상이 발생 될 가능성이 큼

단변량 분석



Originator

1. Originator의 종류별 count를 그래프 표현
2. 결측지 부분만 제거 하고 표현
3. 상위 5개만 표현


- 가장 많이 업무는 System이 4635개 업무 수행
- 그 다음으론 Monica, Dian이 2천여개 업무 수행
- 가장 적은 업무를 맡은 사람의 업무량은 66
- System의 업무량이 앞도적으로 많은것으로 보아 System은 다양한 TaskID를 진행하거나 많은 caseID의 요청이 있을 것으로 보임

이변량 분석

이변량 분석

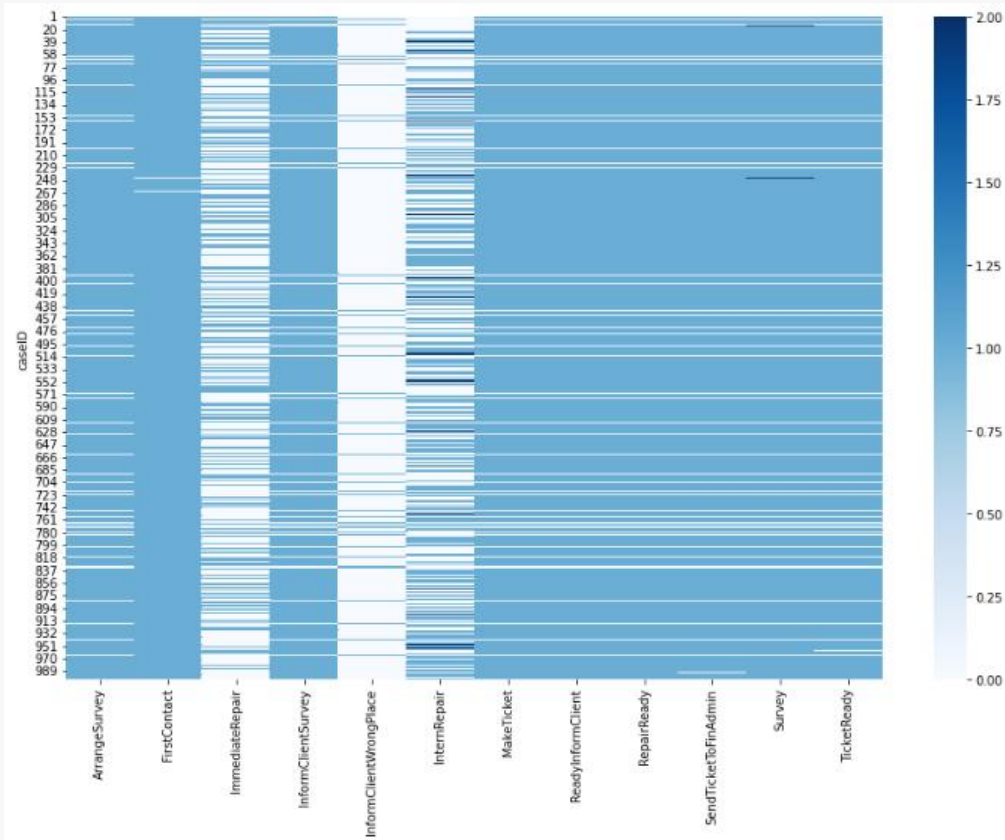


**caseID
&
taskID**



**originator
&
taksID**

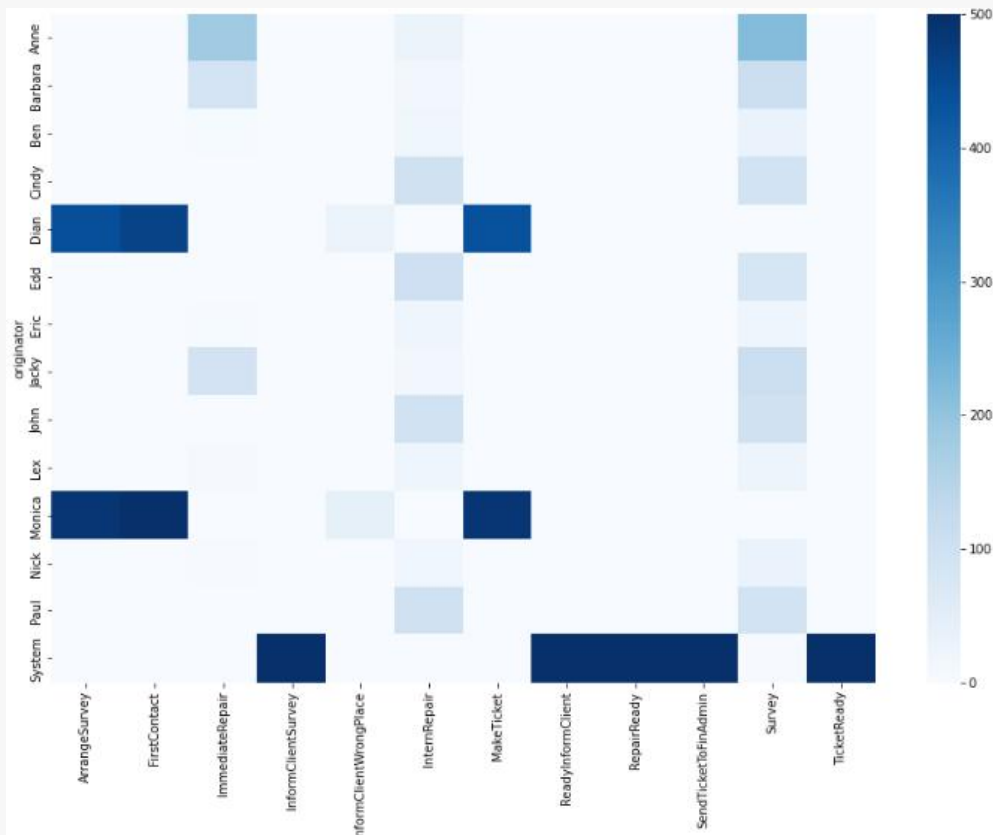
이변량 분석



CaseID & TaskID

1. CaseID 그룹 별 TaskID를 확인하는 그래프
2. immediateRepair와 informClientWrongPlace 가장 적게 이용하는 것을 확인
3. 만약 직원의 배치가 다른 업무보다 많이 배치 되어 있다면 분산하는 것이 좋을 것이라 판단

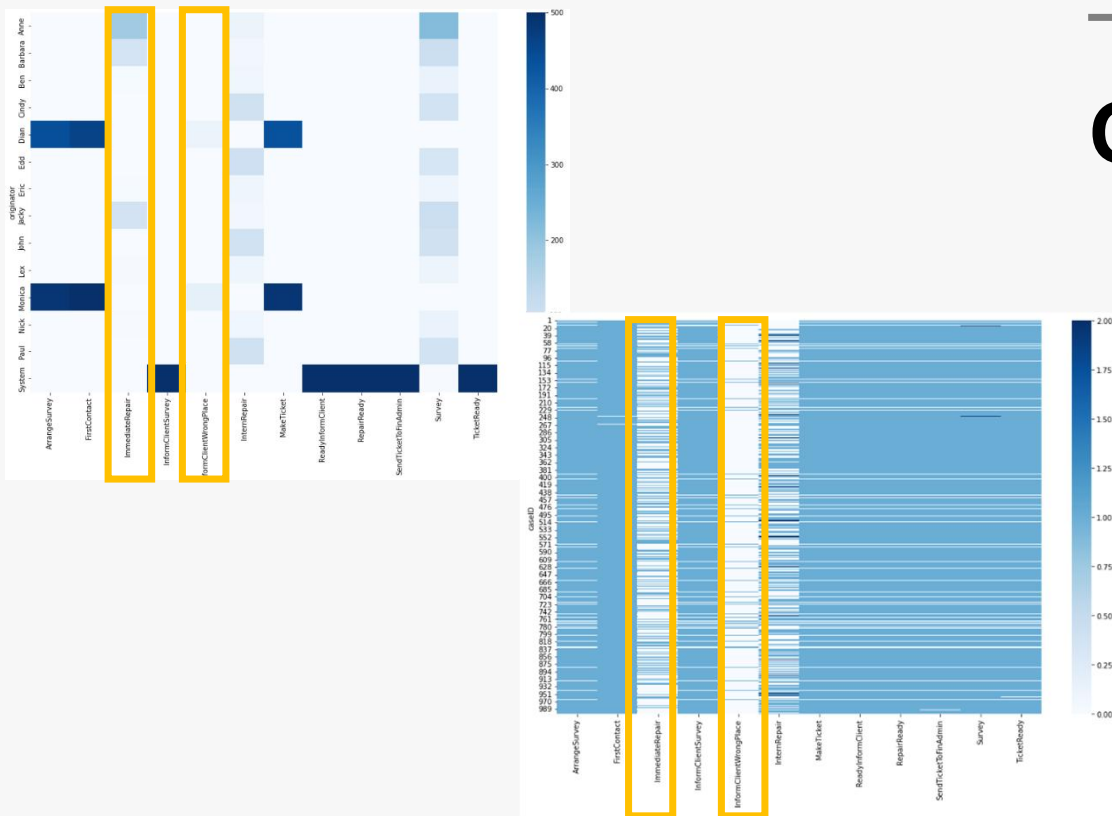
이변량 분석



Originator & TaskID

1. Originator 그룹별 TaskID를 확인하는 그래프
2. System 경우 많은 종류의 task 업무를 맡고 업무량도 많아서 병목현상의 원인이 될 것이라 예상
3. 요청을 적게 받은 업무 경우 3명이나 배치 되어 효율성 떨어짐
4. 반대로 요청을 많게 받던 업무 1명이 집중적으로 담당
5. 만약 그렇다면 직원들을 교육하여 업무를 분산 하거나 직원을 늘리는 것이 해결 가능성이 높음

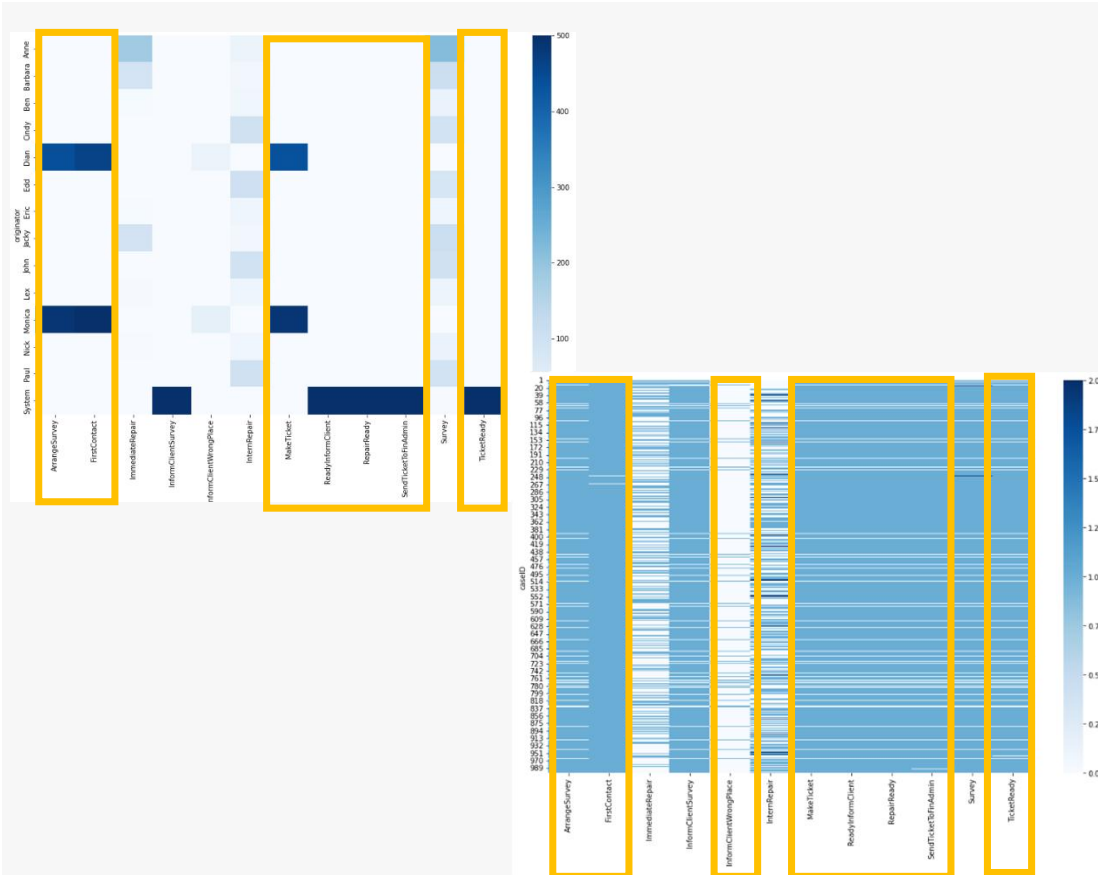
이변량 분석



Originator & TaskID

1. Originator 그룹별 TaskID를 확인하는 그래프
2. System 경우 많은 종류의 task 업무를 맡고 업무량도 많아서 병목현상의 원인이 될 것이라 예상
3. 요청을 적게 받은 업무 경우 2~3명이나 배치 되어 효율성 떨어짐
4. 반대로 요청을 많게 받던 업무 1명이 집중적으로 담당
5. 만약 그렇다면 직원들을 교육하여 업무를 분산 하거나 직원을 늘리는 것이 해결 가능성이 높음

이변량 분석

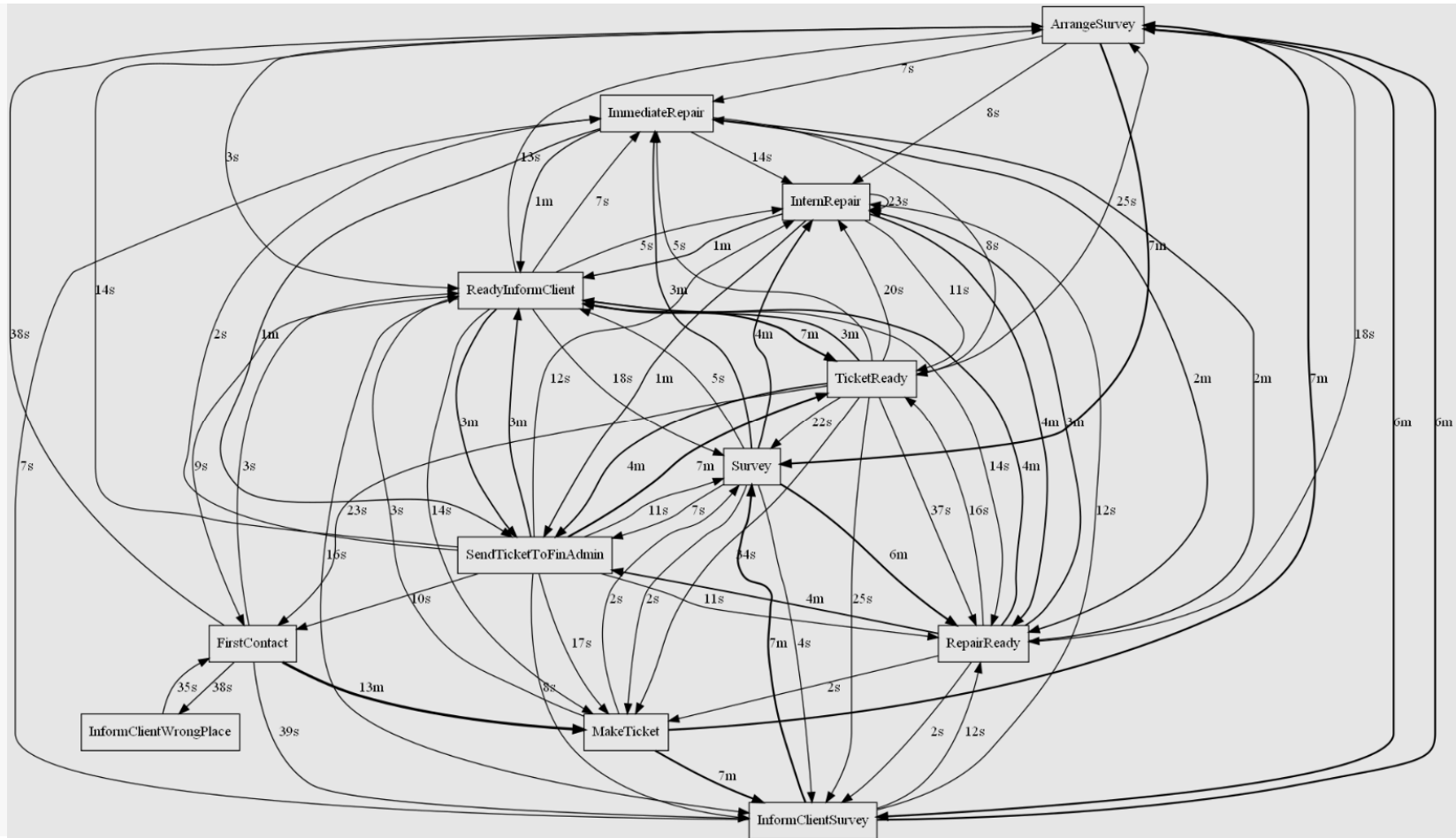


Originator & TaskID

1. Originator 그룹별 TaskID를 확인하는 그래프
2. System 경우 많은 종류의 task 업무를 맡고 업무량도 많아서 병목현상의 원인이 될 것이라 예상
3. 요청을 적게 받은 업무 경우 2~3명이나 배치 되어 효율성 떨어짐
4. 반대로 요청을 많게 받던 업무 1~2명이 집중적으로 담당
5. 직원들을 교육하여 업무를 분산 하거나 직원을 늘리는 것이 해결 가능성이 높음

프로세스 마이닝

프로세스 마이닝

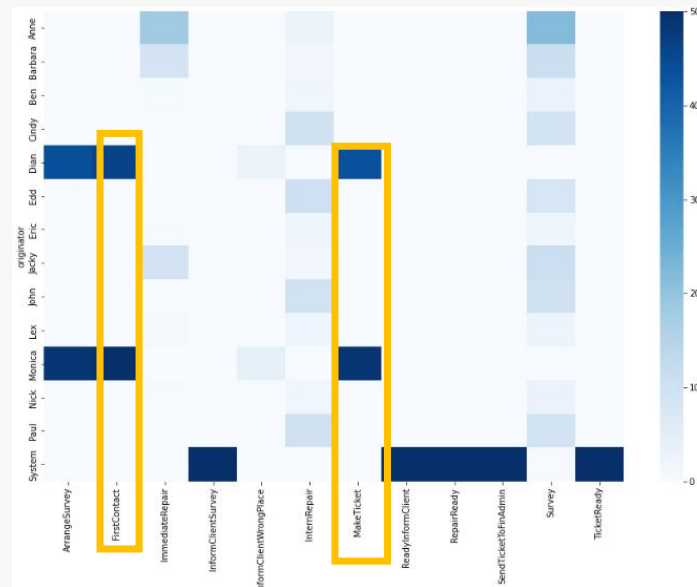


프로세스 마이닝

FirstContact

13m


MakeTicket



직원대비 업무량이 많아 지체 되는 것으로 파악

프로세스 개선 방안

프로세스 개선 방안



인력 확보

인력 분산

1. 업무량에 비해 적은 인력 혹은 많은 인력을 배치하는 문제 발생
-> 적절하게 직원 배치 및 확보
2. firstContact 이후 MakeTicket으로 진행될 때까지의 소요시간이 매우 김
-> 효율적으로 처리할 수 있도록 직원들끼리 새로운 방안을 찾아야할 필요 있음

Thank You

