Chapter 7 Sampling(표본추출) and Sample Distribution

10 마리의 닭을 부위별로 잘라서 동시에 튀기는데 모든 부위가 다 먹기 좋게 익혀있는지 확인을 하기 위해서 잘려진 부위의 개수가 작다면 모든 부위를 검사하는 것도 가능하지만, 10 마리에 10 개 부위가 나온다면 100 조각을 다 검사하는 것은 어렵습니다. 대신 몇 조각만 조사하고 익혀있다고 판단을 하면 기름에서 건저내어서 먹게 됩니다. 이와 같이 100 조각 전체 즉 population 정보를 얻기 위해서 부분만 얻어는 sampling 작업을 통해서 전체를 추측하는 경우가 대부분입니다.

이 처럼 공통된 특징을 갖고 있는 대상을 population 이라고 하는데 여기에 속한 모든 대상을 관찰하기에는 시간과 비용이 많이 들기 때문에 그 중 몇개만 선출하여 sampling 을 실시합니다. 실제 통계 예를 들면 드라마 시청률을 만일 연령별로 조사 한다면 전 국민을 대상으로 조사하는 일은 엄청난 시간과 비용이 발생하게 됩니다. 다른 예로 대통령 선거 기간에 매일 후보자 지지율을 조사하게 되는데 이 또한 전국민 대상으로 조사는 것은 선거날 아니고서야 어려운 일이기 때문에 sample 만 조사를 하여 전체를 대표하는 결과로 사용합니다.

Sampling 은 population 에 대표로 사용되지만 문제가 있습니다. 예를 들어 닭을 부위별로 잘라서 동시에 튀기는데 익혀진 정도를 측정하기 위해서 얇고 작은 부위를 대표로 조사를 하여 다 익혀졌다고 생각하면 두껍고 큰 부위는 익혀있지 않을 가능성이 있습니다. 이러한 것이 잘못된 sampling 에 결과입니다. 실제 예로 대통령 지지도를 분석한다고 sampling 으로 지목된 사람들이 대통령을 강력하게 지지하는 사람들만 선별을 했다면 지지도가 높게 나타나 잘못된 결과를 얻게 됩니다. 이것은 조건 확률에서 말하는 문제와도 유사합니다.

Sampling 다른 문제점이 있습니다. 다음의 기사 내용을 보게 되면 전 GE 회장 잭웰치가 미국 정부가 실업률을 조작하고 있다고 주장을 하면서 조사 당국에 말을 인용을 하며 문제점을 제기했습니다.

잭 웰치, 실업률 조작 또 주장

미국 정부의 '실업률 조작' 주장으로 논란을 일으킨 잭 웰치 전 제너럴일렉트릭(GE) 회장이 또다시 버락 오바마 행정부가 실업률 수치를 조작했다는 칼럼을 게재했다.

웰치 전 회장은 9 일(현지시간) 월스트리트저널(WSJ) 온라인판에서 근거를 여럿 제시하며 지난 9 월 실업률이 "매우 잘못됐다"는 주장을 폈다. 그는 앞서 자신의 트위터를 통해 9 월 미 실업률이 7.8%에 그쳤다는 노동통계국(BLS)의 최근 발표에 대해 조작 의혹을 제기했었다.

그는 WSJ 온라인판에 실린 '내 주장이 옳았다(I Was Right About That Strange Report)'라는 제목의 칼럼에서 BLS 의 실업률 조사 시스템의 한계를 지적하며 당국의 조사방법이 객관적이고 정확하다는 것은 "과장"이라고 비판했다. 그는 실업률이 매달 일주일 정도 전화나 가정방문을 통해 조사되는데 조사당국인 BLS 도 이 방법과 관련, "응답자가 질문 내용을 잘못 이해하거나 빠진 자료를 추산하는 과정에서 오류가 발생할 가능성이 있다"며 한계를 인정했다고 밝혔다.

서울경제 2012/10/10

내용에 보듯이 sample 로 선정된 응답자로 부터 얻는 자료에서 응답자의 질문에 대한 이해 부족으로 인해 잘못된 결과가 나타날 수 있다는 것을 지적하고 있습니다.

이처럼 population 전체를 이해하기 위해서 선별된 sample 들의 타당성 문제, sample 로부터 얻은 자료가 정확하지 못하거나 자료를 취합하는 과정에서 오류와 같은 문제는 매우 조심해야할 것들입니다. 이러한 실수로 잘못된 결과를 적용하여 사업을 한다면 기업에 막대한 손실을 줄 수 있습니다.

1. Sampling 유형

1.1. 확률 표본 추출방법

1) Simple Random Sampling(단순 무작위 표본추출)

Population 에 모든 data 는 선택될 확률이 같은 uniform distribution 과 같은 상황에서 랜덤으로 추출을 합니다.

2) Systematic Sampling(체계적 표본추출)

Population 에 data 를 선별은 어느 시작점으로 부터 몇 번째 값을 추출하는 방식으로 예를 들어 음료수 공장에서 1000 번째 생산된 음료를 검사하는 것입니다. 이러한 방식에 문제가 될 수 있는 것은 만일 음식점 매출에 대한 조사를 위해서 매주 화요일을 검사하는 것과 매주 일요일을 검사하는 것은 다른 결과가 나오게 되기 때문에 전체를 대변하기에는 어렵습니다.

3) Stratified Sampling(층화 표본추출)

Population 에 data 를 group 별로 나누어 group 별 비율에 맞도록 sample 을 추출하는 것으로 예를 들어 서점의 회원이 총 1000 명이 있는데 이 회원들 중에 모바일 서비스에 선호도를 조사를 하려고 합니다.

연령대	시청자수	비율
10 대	400	40%
20 대	300	30%
30 대	200	20%
40 대	100	10%

이러한 상황에서 만일 선별을 하였는데 비율이 실제 비율과 반대로 나와 40 대가 40 명, 30 대가 30 명, 20 대가 20 명, 10 대가 10 명이 선출이 되었다면 선호도 조사 결과는 회원 비중이 총 30%만 되는 40 대와 30 대에 의해서 결정이 됩니다. 만약 10 대와 20 대는 만족을 하지만 30 대와 40 대가 만족하지 않는다면 결과는 만족하지 않는다쪽으로 결론이 날 수도 있게 됩니다.
이러한 것을 방지하기 위해서 sampling 을 총 비율에 맞도록 하는 것이 stratified sampling 입니다. 예를 들어 모바일 서비스 선호도 조사에서 100 명을 선출하려고 할 때 10 대에서 40 명, 20 대에서 30 명, 30 대에서 20 명, 40 대에서 10 명을 선출을 random sampling 이나 systematic sampling 으로하면 됩니다.
이와 같이 Stratified sampling 은 상호 배제되는 group 별로 생각이 다르기 때문에 올바른 대표들을 선출하는 sampling 에서 group 에 비율에 맞도록 선출하는 것입니다.

4) Cluster Sampling(군집표본추출)

Stratified sampling 과 같이 상호 배제되는 group 별로 구분이 되는 population 로 부터 sample 을 선출하는 방식입니다. 차이점은 stratified sampling 에 group 은 유사한 특성을 갖고 있지만 cluster sampling 에 group 은 유사성이 없을 수 있는 여러 특성이 혼합된 특성을 갖고 있습니다.
예를 들어 지역을 나누어서 sampling 을 하는 경우와 같이 지역을 population 을 묶는 단위로 하여 그 내부에서 sample 들을 선별합니다. 예로 서비스 센터의 고객만족도를 지역별로 알아 보려고 할 때 해당 지역의 고객 만족도는 연령별, 성별등 다양한 특성들이 혼합이 되어 있습니다.

5) Resampling : boostrapping

Population 으로 부터 sampling 을 여러번 하여 결과를 얻는 방법입니다. 이 기술이 유용한 때는 mean, standard deviation 과 같은 값을 계산 할 때 입니다. 여러번 sampling 을 통해서 mean 을 계속 찾다보면 population 에 mean 에 가까워 지는 원리를 이용한 것입니다.

jMath 를 이용해서 sampling 에 사용될 sample 의 index 값들을 구할 수 있습니다.

jMath.sampling(numsamples, numpops, type, opts)

입력값	설명
numsamples	Sample 로 사용될 총 개수
numpops	Sample 로 사용될 수 있는 총 개수. * Type 이 stratified 인 경우 각 group 별 총 개수들을 나열한 배열.
type	'random': simple random sampling(단순 무작위 표본추출) : 기본값 'systematic': systematic sampling(체계적 표본추출) 'stratified': stratified sampling(층화 표본추출)
opts	Type 마다 다른 option 값들 'systematic': 표본 추출 간격 'stratified': 각 group 별 선택 비율

Simple random sampling : 1000 개중에 10 개만 무작위로 sample 을 선출합니다.

```
> jMath.sampling( 10, 1000, 'random')
[634, 941, 99, 223, 9, 501, 202, 132, 459, 77]
```

Systematic sampling: 1000 중 시작점을 하나 잡고 30 간격으로 sample 을 선출합니다.

```
> jMath.sampling( 10, 1000, 'systematic', 30)
[135, 165, 195, 225, 255, 285, 315, 345, 375, 405]
```

Stratified sampling: 1000 중 4 개의 group 으로 나누어 비중을 각각 40%, 20%, 10%, 30%로 sample 을 선출합니다.

```
> jMath.sampling( 10, [1000, 300,200,100], 'stratified', [0.4,0.2,0.1,0.3])
[[ 141, 391, 18, 137], [ 43, 30], [6], [4, 7, 16]]
```

1.2. 비확률 추출 방법

이 방식들의 특징은 비용과 시간이 확률방식 보다는 작지만 선출된 sample 이 population 을 대표한다고 판단하기는 어렵습니다.

1) Convenience sampling(편의 표본추출)

앞의 sampling 은 공통된 특징인 있는 사람들(population)을 대상으로 random 으로 확률을 이용해서 조사 대상을 추출하지만 convenience sampling 의 경우는 조사 대상에 특징이 없습니다. 예를 들어 신제품 만들고 지나가는 행인 아무한테나 주고 반응을 보는 것입니다. 즉 주먹구비식으로 결론을 만들어 나가는 것을 말합니다. 이러한 방식은 아무리 sample 로 사용된 자료가 많다하여도 정확한 결과를 얻기 힘듬니다.

2) Judgement sampling(판단표본추출)

조사자가 판단하여 표본 추출하는 방식으로 모든 것은 조사자의 판단에 의존하기 때문에 대표성이 높다고 할 수 없다.

3) Quota sampling(할당표본추출)

Stratified sampling 와 convenience sampling 을 합친 방법으로 group 별로 선출되는 인원은 다르지만 그 안에서 뽑는 방식은 조사자가 편리한 장소와 시간에 접속하기 쉬운 대상을 선출하는 방식입니다.

4) Snowball sampling(누적표본추출)

조사자가 임의로 선정한 sample 에 있는 사람들로 부터 다른 사람들을 소개 받아서 sample 을 키워가는 것을 말합니다.

2. Sampling 과 nonsampling 오류

Data 로 부터 정보(information)를 만드는 작업을 하는 Statistics 에서 정보를 나타내는 값으로 평균, 표준편차, median, mode 같은 것이 있는데 이것이 population 으로 부터 얻은 것이라면 parameter 들이라 합니다. 반면 sample 로 부터 얻은 값이라면 statistic 들이라고 합니다.

하지만 sample 로 부터 얻은 값은 population 에서 얻은 값과 같이 않을 경우가 대 다수입니다. 이러한 값의 차를 sampling error 라고 합니다. 평균 값에 대한 sampling error 는 다음과 같이 표기 됩니다.

$$\text{Sampling Error} = \text{statistic} - \text{paramter}$$
$$\text{평균 Error} = \bar{x} - \mu \tag{7.2}$$

Sampling error 를 줄인다고 무작정 sampling 크기를 늘리는 것은 아무런 효과가 없으니 sampling 을 잘 해야하는 것이 중요합니다. 또한 수 많은 데이타로 구성된 population 에서 평균값과 같은 정보를 얻는 것은 어렵기 때문에 sampling error 를 정확하게 알기에는 힘들지만 확률로 값의 정확도를 어느 정도 예측을 할 수 있습니다. 다시말해, sample 로 부터 얻은 statistic 으로 parameter 값이 존재할 범위를 찾을 수 있습니다.

이러한 오류 외로 sampling 과정에서 발생하는 오류가 있습니다. 이것을 nonsampling error 라고 하는데 예를 들어 측정되는 결과가 부정확하거나 설문 응답자가 이해를 잘못해서 잘못된 응답을 할 경우들이 있습니다. 즉 잘못된 sampling 에 의한 오류보다는 sample 들로 부터 측정하려는 정보의 부정확으로 발생되는 오류를 말합니다. 다른 오류로 질문을 어떻게 하는가에 따라 응답이 다르게 나오는 경우가 있습니다. 다음의 신문기사를 보겠습니다.

'장기기증 동의'미국 15%, 유럽선 90% 왜?

묻는 방식따라 답변 하늘과 땅 차이

운전면허증을 찬찬히 들여다보라. ID 오른쪽 하단에 조그만 핑크색 동그라미가 하나 있을 것이다. 동그라미는'최악의 사태가 발생할 경우' 자신의 장기를 기증할 것인지 아닌지에 관한 면허증 소지자의 결정을 보여준다. 동그라미 안에 기증자(donor)라는 단어가 적혀 있으면'내가 사고사를

당할 경우 몸 안의 장기를 적출해도 좋다'는 뜻이다. 반면 빈 동그라미면 기증의사가 없다는 뜻이다.

미국선'동의하나' 질문, 특별한 희생 여부 결정
유럽은'동의하지 않나' 물어봐 의무감을 강조

인명피해를 수반한 교통사고가 발생하면 경찰을 비롯한 1 차 대응팀은 부상자와 사망자의 신원을 파악하기 위해 면허증을 확인하는데 이 과정에서 자연스레 장기기증에 관한 이들의 결정을 알게 된다.

그런데 한 가지 흥미로운 점은 장기기증 여부를 묻는 방식을 달리하면 각 개인의 결정에 큰 영향을 끼치게 된다는 사실이다.

미국의 모든 운전자들은 면허증 신청서 작성 때 장기기증 의사를 묻는 항목에 '예' '아니오'로 자신의 의사를 분명히 밝혀야 한다.

신청서의 해당항목의 문구는 '(내게) 최악의 사태가 발생할 경우 장기 기증에 동의한다'로 되어 있다.

반면 유럽의 일부 국가들 '장기기증에 동의하지 않는다'는 문안을 제시한 후 개인의 선택을 요구한다.

이때 '기증에 동의한다'는 긍정적 서술에 대해 '네' '아니오'의 답변을 요구하는 것을 '옵트-인'(opt-in) 방식이라 부르고 그 반대, 즉 '동의하지 않는다'는 문안에 대한 답변을 요구하는 것을 '옵트-아웃'(opt-out) 방식이라 부른다.

다시 말해 옵트-인 방식에서는 각자가 기증자가 될 것인지를 선택해야 하는 반면 옵트-아웃 방식에서는 기증자가 되지 않을 것인지를 결정해야 한다.

결국 똑같은 것 아닌가 싶겠지만 전혀 그렇지 않다. 다른 무엇보다 결과에서 큰 차이가 난다.

미국 등 옵트-인 방식을 택한 국가들의 경우 전체 운전면허 신청자의 15%만이 신장 기증을 택하는데 비해 옵트-아웃 프로그램을 시행하는 곳에서는 90% 이상이 기증을 결정한다.

한국일보 2012-11-20

기사 내용은 framing effect 라하여 같은 내용을 다른게 표현을 하여 다른 감정을 이끌어 내는 것으로, 예를 들어 98%의 안전률과 2%의 사고률은 같지만 다른 느낌을 전달합니다. 동의를 할 경우에 체크를 하는 opt-in 방식의 경우는 장기 기증을 미리 생각한 사람들에게 바로 응답을 하는 것이지만 그렇지 않는 경우는 한번 더 생각을 하게 만들기 때문에 동의률이 떨어질 수 밖에 없습니다.

마지막으로 Sample 의 크기가 population 의 크기와 같다면 이것을 census(총조사)라고 합니다. 이경우에 평균은 population 평균이기 때문에 sample error 는 0 이 됩니다.

3. The Central Limit Theorem(CLT)

이론이란 말 때문에 어려워 보일 수 있지만, 매우 단순한 개념입니다. 이것이 중요한 이유는 앞서 설명드린 sample error 로 parameter 의 정보를 정확하게 얻지 못하는 것을 찾을 수 있도록 해주기 때문입니다.

예를 들어 만개의 커피 원두가 담겨 있는 봉지가 있는데 모든 봉지에 무게가 1kg 이라고 할 때 sample 들을 30 개씩 선별을 해서 30 개에 대한 평균 값(sample mean)을 얻어 보았을 때 다음과 같은 결과를 얻었습니다.

무게(kg)							Sample 평균
0.9953	0.9973	1.0110	0.9972	1.0070	0.9795	0.9965 …	1.001
0.9990	0.9761	1.0306	0.9960	0.9786	1.0405	0.9933 …	0.9823
0.9918	0.9842	1.0051	1.0028	1.0003	0.9867	1.0113 …	0.9912
1.0035	0.9970	1.0002	0.9974	0.9825	0.9971	0.9917 …	1.0158
0.9804	0.9769	0.9893	0.9599	1.0193	1.0104	0.9996 …	0.9974
0.9804	0.9980	0.9879	1.0291	1.0083	1.0138	0.9894 …	0.9812

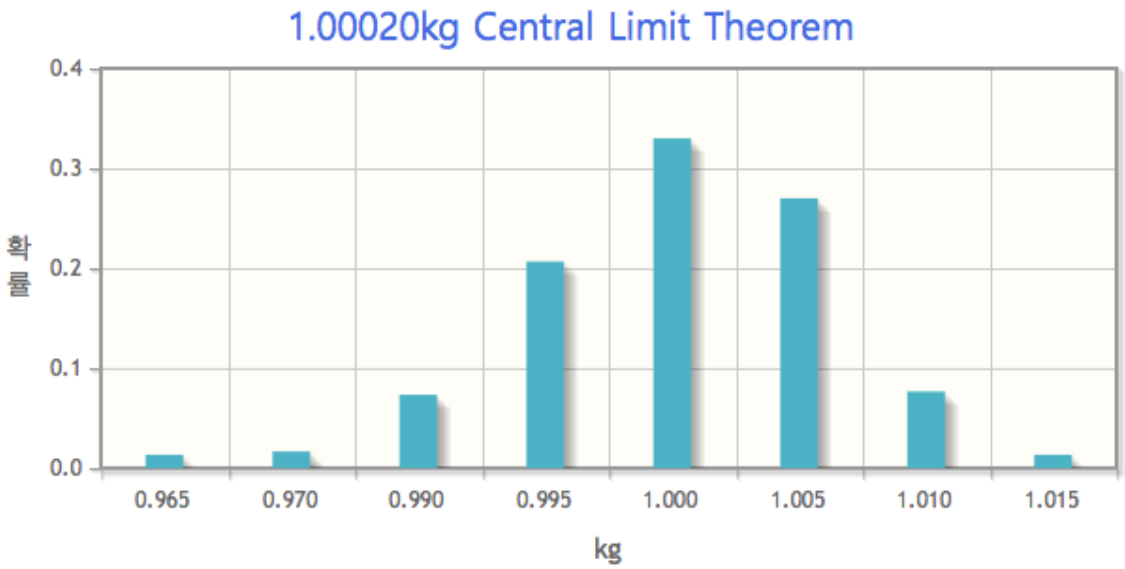
그런데 오른쪽에 있는 sample 평균들의 분포를 구해보면 normal distribution 의 모양이 됩니다. 이것이 Centrol Limit Theorem 의 핵심적인 내용입니다.

이해를 돕기 위해서 simulation 을 해보도록 하겠습니다. 0.95kg 에서 1.05kg 까지 만개의 봉지를 uniform distribution 으로 생성을 합니다. 그런 다음 30 개씩 sample 을 random 으로 추출해서 평균을 구하고 이 평균값을 0.005kg 단위로 모아서 histogram 을 그려 보겠습니다. 수식 (7.3)을 적용한 sampling distribution 을 그려 보도록 하겠습니다.

$$\mu_{\overline{x}} = \frac{\sum_{i=1}^{M} \overline{x}_i}{M} \tag{7.3}$$

여기서, $\bar{x}_i$는 i 번째 sample 들에 평균 값이고 M 은 총 sampling 한 횟수입니다.

chapter07/7_coffee.html



Uniform distribution 으로 되어 있기 때문에 무게가 random 으로 0.95kg 에서 1.05kg 사이로 퍼져 있어 평균은 수식 6.7 에 의하여 1kg 이 됩니다. 그런데 실험을 한 결과를 보시면 30 개씩 모아서 평균을 냈을 때는 평균값이 0.995kg ~ 1.005kg 사이가 가장 많이 발생 되고 이것을 기준으로 양쪽으로 대칭적으로 퍼져 있는 normal distribution 을 형성하는 것을 볼 수 있습니다. 만개의 봉지에 무개에 실제 평균을 보시면 1kg 에 가깝다는 것을 알수가 있어 sample mean 을 중심으로 error 반경 어느 정도에 population 에 해당하는 평균이 존재함을 알 수 있습니다.

이 처럼 Central Limit Theorem 이 말하는 것이 여러번 측정하여 측정된 값들에 평균을 모으면 Population 에 해당하는 평균값 주변에 sample 평균값이 많이 분되어 population mean 이 이 영역안에 존재함을 추론할 수 있습니다. 이러한 특성 때문에 normal distribution 은 매우 중요한 도구가 되고 sample mean 을 이용해서 population mean 이 있을 수 있는 범위를 말할 수 있게 됩니다. 예를 들어 한번의 sampling 으로 얻은 sample 들로 부터 sample mean 을 구한 결과 0.98kg 이 나온다면 이 값 주변에 population mean 이 있을 확률을 normal distribution 을 이용하여 얻게 됩니다.

Simulation 코드

https://github.com/handuck/jMath

```
// 만개의 커피 봉지 무게를 0.95 ~ 1.05kg 이 되도록 생성한다.
var coffeeweights = [];
for ( var i = 0 ; i < 10000 ; i++ )
{
     var p = Math.random() * 0.1 + 0.95;
     coffeeweights.push( parseFloat(p.toFixed(3)) );
}

// 만개의 커피 봉지에서 numSamples 만 simple random sampling 으로 선출해서
sample 평균을 계산합니다.
function averageWeight(numSamples)
{
     var p = jMath.sampling(numSamples, coffeeweights.length )
       .reduce( function(acc,value,index) {
              return coffeeweights[value] + acc;
       },0)/numSamples;
     return parseFloat(p.toFixed(3));
}

// Resampling 을 300 번 실시 합니다.
var list = [];
for ( var i = 0 ; i < 300 ; i++ )
{
    list.push( averageWeight(30) );
}

// 만개 커피 봉지의 무게 평균을 계산합니다.
var mu = coffeeweights.reduce( function(acc,value){
              return acc + value;
              },0) / coffeeweights.length;
```

그럼 sample mean 들에 standard deviation 값인 sample 평균의 standard error 를 알아보겠습니다. 이 계산을 위해 우리는 population 에 standard deviation 값 σ을 알고 있다고 할 때 값은 다음과 같습니다.

$$\sigma_{\bar{x}}^2 = \frac{1}{M}\sum_{j=1}^{M}(\bar{x}_j - \mu)^2 = \frac{1}{n}\frac{1}{nM}\sum_{j=1}^{M}\left(\sum_{i=1}^{n}(x_{ij} - \mu)\right)^2 = \frac{1}{n}(E(x^2) - \mu^2) = \frac{\sigma^2}{n} \qquad (7.4)$$

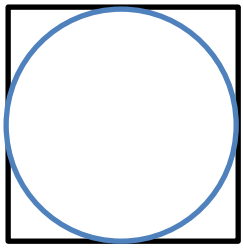
여기서 n 은 한 번 sample 할 때 크기이고 M 은 이러한 총 sampling 작업을 한 횟수 입니다.

수식 7.4 가 말해주는 것은 sample 할 때 마다의 개수가 많을 수록 sampling 의 평균간의 간격은 좁혀진다는 것으로 다시 말해, 큰 sample 크기를 사용할 수록 population 에 가까운 평균값을 얻을 수 있다는 의미고 다른 것은 실제 population standard deviation 이 작다면

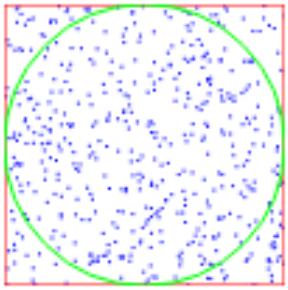
sampling 으로 얻은 평균들에 standard deviation 에 평균 또한 좁게 분포가 된다는 뜻입니다.

CLT 을 정리하면 비록 population 이 normal distribution 이 아니라도 sample 들의 평균값들의 분포는 normal distribution 이기 때문에 normal distribution 으로 평균값을 예측할 수 있습니다.
다른 중요한 것으로 만일 population 이 정말 normal distribution 이면 sample 에 크기에 상관없이 sample 평균들에 분포는 normal distribution 을 형성하지만 그렇지 않을 경우 측정의 위한 한 sample 의 크기는 30 보다 커야 sample 평균의 분포가 normal distribution 이 될 수 있습니다. 만일 normal distribution 일 경우에는 sample 크기가 30 보다 작아도 상관이 없습니다.

다른 예를 들면 JavaScript 으로 원주율값 3.141592....를 계산을 sampling 을 통해서 할 경우를 소개 하겠습니다. 방법은 가로와 세로 길이가 1 인 정사각형 안에 원을 넣으면 넓이는 0.25π가 됩니다.



그런데 이러한 원의 넓이를 구하는 다른 방법은 사각형에 점을 찍어서 점이 원내부에 있는 비율을 구하게 되면 원의 넓이가 된다는 것이고 이 비율을 p 라고 한다면 원주율 π값은 4p 가 됩니다.



원안에 점을 찍는 것은 Math.random 을 두번 실행해서 각각을 x,y 로 하고 다음에 수식을 만족을 하면 원내부 그렇지 않으면 원 외부로 인식하면 됩니다.

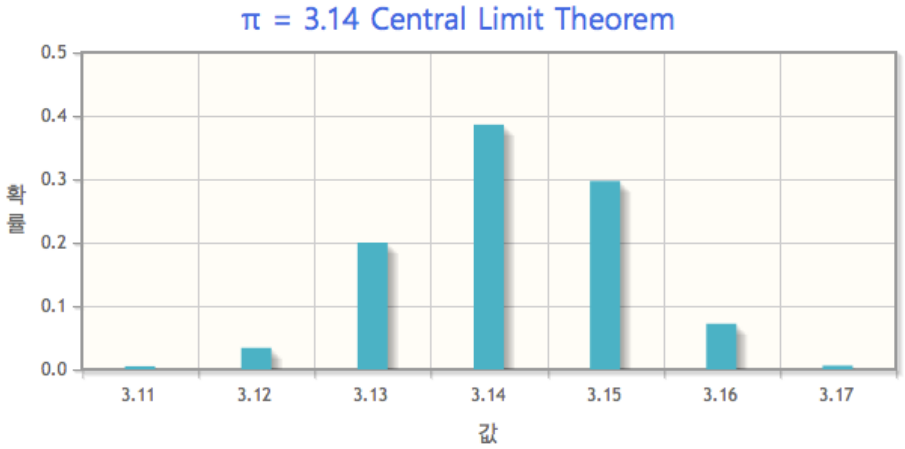
$$(x - x_c)^2 + (y - y_c)^2 \leq r^2$$

원주율 π를 계산하는 JavaScript code 는 다음과 같습니다.

```
cnt  = 0;
for( var i = 0 ; i < len ; i++ )
{
    var x = Math.random();
    var y = Math.random();
    cnt += (Math.pow(x - 0.5, 2) + Math.pow(y - 0.5, 2)) <= 0.25 ? 1 : 0;
}
cnt / len / 0.25
```

이러한 방식으로 점을 여러번 찍어서 비율을 계산하여 얻은 원주율 값을 한 sample 로 보고 이 것을 30 번 실행하여 얻은 값이 sample mean 이 됩니다. 그럼 이 sample mean 여러번 계산을 하여 histogram 을 보시면 다음같은 원주율 분포가 나타납니다.

chapter07/7_pi.html



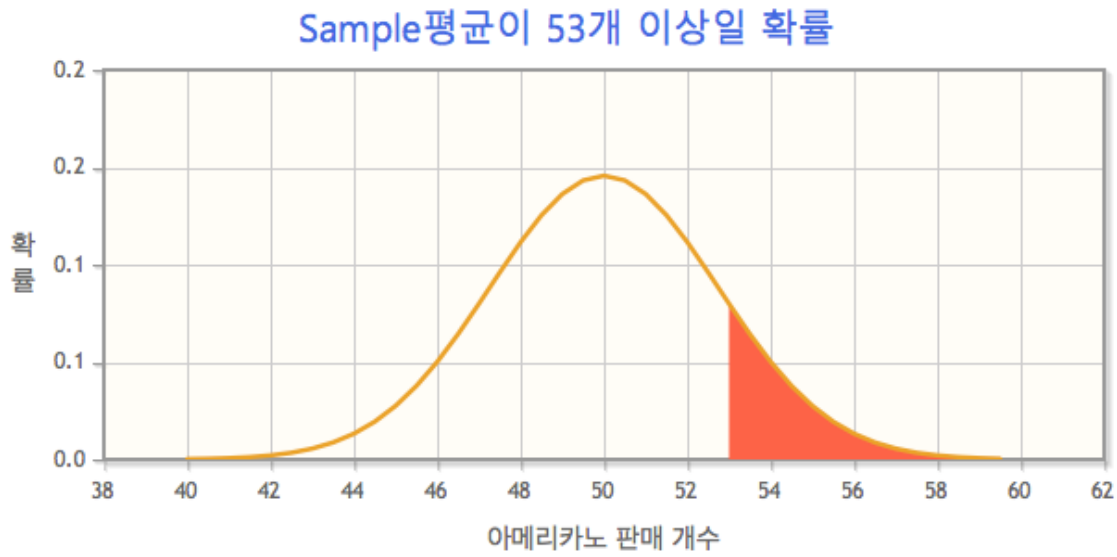
모든 sample 값인 4p 값을 30 개씩 묶어 sample 값에 평균을 갖고 분포도를 보면 normal distribution 형태로 나타남을 알 수 있습니다. 이것이 CLT 가 말하는 것입니다.

CLT 이 normal distribution 으로 형성되는 sample mean 들의 분포를 형성을 말해 주기 때문에 z-score 를 이용할 수 있습니다.

$$z_{\overline{x}} = \frac{\overline{x} - \mu_{\overline{x}}}{\sigma_{\overline{x}}} \quad (7.5)$$

즉, Normal distribution 으로 확률을 계산하는 것을 할 수 있게 됩니다. 예를 들어 커피전문점에서 주중 점심 시간 12 시 부터 1 시 사이에 판매되는 아메리카노의 개수를 측정을 하려고 할 때, 한 sample 의 크기를 30 개씩 해서 sample 평균에 평균을 구한 값이 50 개이고 전체 population 에 표준편차가 15 개라고 했을 경우 sample 평균이 53 개 이상일 확률을 구하려면 다음과 같습니다.

chapter07/7_clt_coffee.html



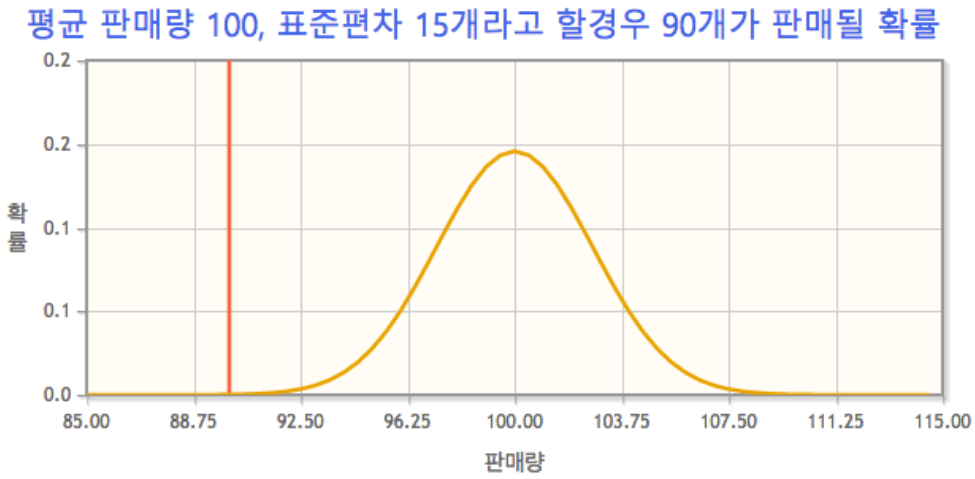

```
> 1 - jMath.stat.normcdf( 53, 50, 15/Math.sqrt(30) )
0.13666083914614902
```

즉 하루 동안 판매된 아메리카노의 평균이 53 개 이상이 될 확률은 13.6%이 됩니다. 이러한 결과를 얻을 있는 이유는 population 데이타 분포 유형이 어떻든 상관없이 CLT 덕분에 sample 평균의 평균 분포가 normal distribution 이기 때문입니다.

CLT 를 이용해서 주장하는 값으로 부터 정확도를 알 수 있습니다. 예를 들어 비싼 메뉴를 내놓은 1500 개의 프렌차이 점을 갖고 있는 프렌차이 회사에서 매장에 하루 평균 판매되는 양이 100 개라고 광고를 합니다. 그런데 운영이 어느 정도 잘된다는 프렌차이점들을 30 군대 조사를 해보니 하루 평균 판매량이 90 개로 나나타고, 업체가 주장한 Population 에 표준편차를 15 개라고 했을 때 100 개가 타당한가를 알아 보면 다음과 같습니다.

chapter07/7_clt_franchise.html

https://github.com/handuck/jMath



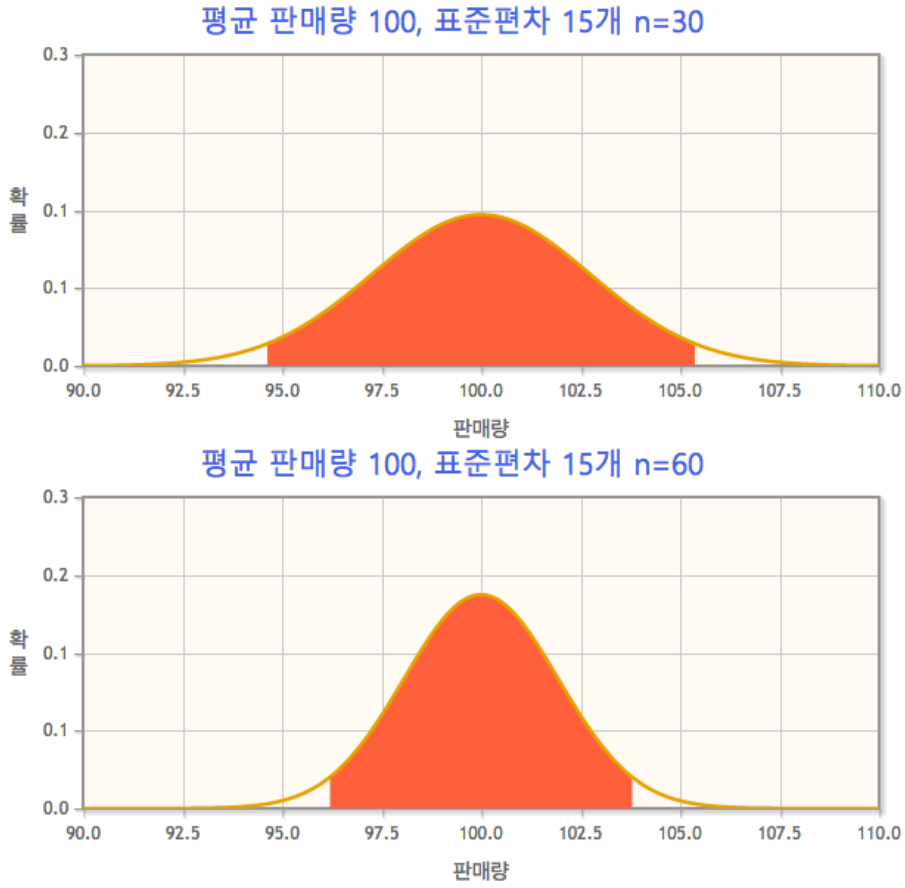

```
> jMath.zscore( 90, 100, 15/Math.sqrt(30))
-3.6514837167011076

> jMath.stat.normcdf( 90, 100, 15/Math.sqrt(30));
0.00013036481642764164
```

결과에 보듯이 100 라는 가정하게 90 개가 나올 확률은 0.00013 으로 매우 희박함을 알 수 있고 z-score 만 보아도 -3.65 는 95%범위인 2.5% ~ 97.5%범위에 해당하는 z-Score -1.96 ~ 1.96 밖으로 만일 정말 매장당 하루 평균 판매량이 100 개라면 90 개가 나타날 확률은 매우 희박하다는 말이기 때문에 하루 평균 판매량이 100 개라는 것은 의심을 해 봐야 합니다. 그럼 회사가 주장을 어느 정도 지지하기 위해서는 30 가계의 평균 판매량이 95% 범위에 있어야 한다면 평균 판매량은 다음과 같아야 합니다.

$$\overline{x}_U = \mu_{\overline{x}} + z_{0.975}\sigma_{\overline{x}}$$
$$\overline{x}_L = \mu_{\overline{x}} + z_{0.025}\sigma_{\overline{x}}$$

chapter07/7_clt_samplesize.html




```
> 100 + jMath.stat.norminv(0.025,0,1) * 15/Math.sqrt(30)
94.63241756884852

> 100 + jMath.stat.norminv(0.975,0,1) * 15/Math.sqrt(30)
105.36758243115148
```

대략적으로 94.6 ~ 105.4 개 사이에 값이 나와온다면 sample error 로 관주하여 어느 정도 회사가 주장하는 것이 맞을 수 있습니다.

여기서 주목할 점은 바로 sample 크기가 늘어나면 sample error 가 줄면서 평균 분포가 좁아지기 때문에 따라서 95%의 폭도 줄어 든다는 것입니다.

```
> 100 + jMath.stat.norminv(0.025,0,1) * 15/Math.sqrt(60)
96.20454606435501

> 100 + jMath.stat.norminv(0.975,0,1) * 15/Math.sqrt(60)
103.79545393564499
```

Sample 평균을 구하기 위해서 가계수를 30 에서 2 배 늘린 결과 95%의 지지를 할 sample 평균값의 범위는 양쪽으로 줄어들어 96.2 ~ 103.8 이 됩니다.

다음은 신뢰성 관련하여 자동차 업체가 연비 과장으로 소송을 건 신문 기사 내용입니다.

1700 명 '연비 소송'..이기면 30 억 보상, 승소 가능성은?

자동차 소유주 1700 여명, 국내외 자동차업체 6 곳 상대 손해배상 소송 제기

부풀려진 자동차 연비로 피해를 입은 소비자들이 국내외 자동차회사들을 상대로 법정 다툼을 시작했다. 10 년동안 추가로 지출되는 유류비와 '뻥튀기 연비'로 부풀려진 차값을 돌려달라는 주장이다. 이번 소송에서 이길 경우 이들이 받을 수 있는 보상금 규모는 30 억여원에 이르러 소송 결과가 주목된다.

법무법인 예율은 7 일 자동차 소유주 1785 명을 대리해 현대자동차, 쌍용자동차 등 국내외 자동차 제조업체 6 곳을 상대로 서울중앙지법에 손해배상 청구소송을 제기했다.

구체적으로는 현대차 싼타페 DM R2.0 2WD 소유자 1517 명이 150 만원, 쌍용차 코란도스포츠 CW7 4WD 소유자 234 명이 250 만원씩을 각각 청구했다.

외제차의 경우 BMW 미니쿠퍼 D 컨트리맨 소유자 7 명이 90 만원, 크라이슬러 지프 그랜드체로키 2013 소유자 3 명이 300 만원, 아우디 A4 2.0TDI 소유자 6 명이 90 만원, 폭스바겐 티구안 2.0TDI 소유자 18 명이 90 만원씩을 청구 금액으로 제시했다.

이번 소송은 지난 2 월 국토교통부가 시중 차량의 연비 검증 결과 6 개 차량의 표시연비가 법에서 허용한 오차 5%를 크게 벗어났다는 부적합 판정 결과에 따른 것이다. 소비자들의 집단 연비소송은 지난해 1 월 현대차와 기아차를 상대로 한 소송에 이어 이번이 두 번째다.

머니투데이 2014/07/07

기사에내용을 보면 허용한 오차 범위 5%를 크게 벗어났다고 합니다. 이러한 검사를 앞서 소개드린 방식으로 하였다면 회사들이 제시한 연비는 population 평균값으로 하여 국토부에서 차량을 검사 한 sample 평균과 회사에서 제시한 표준편차에 적용을 하여 확률을 계산한 결과 95%영역 밖에 sample 평균이 있기 때문에 부적합 판정을 내린 것입니다.

- **Population 이 크기가 크지 않을 경우**

N 이 population 의 크기이고 n 이 sample mean 을 구하기 위한 크기라고 했을 때 N 이 크면 n/N 의 비율은 0 에 가깝지만 만일 n/N 비율이 5%이상일 경우 수식 7.4 에 평균의 standard error 에 수정이 필요합니다.

$$\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{n}}\sqrt{\frac{N-n}{N-1}} \tag{7.6}$$

여기서 $\sqrt{\frac{N-n}{N-1}}$을 finite population correction factor 라고 합니다.

예를 들어 한 버스 회사에서 디젤 버스가 100 대가 있는데 NOx (질소 산화물)의 배출량을 자체 조사 하였는데 평균 6 g/km 로 조사가 되고 표준 편차가 2 g/km 라고 발표를 했습니다. 그런데 이 중 30 개 버스를 조사를 해보니 평균 6.7 g/km 로 나타났을 경우 회사 측이 조사된 값이 정확한지를 알아 보는 방법으로 z score 를 조사하면 되는데 여기서 population 이 100 개이고 sample 로 30 개를 조사 했기 때문에 총 30%의 조사는 수식 7.6 으로 평균의 standard error 로 계산을 해야 합니다.

```
> sigma = 2/Math.sqrt(30) * Math.sqrt( 70/99 )
0.30704412431455885

// z-score
> (6.7 - 6)/sigma
2.2798026230356

> 1 - jMath.stat.normcdf( 6.7, 6, sigma )
0.01130969868766174
```

결과를 보시면 z-score 는 2.28 로 6 g/km 가 평균일 때 95%범위 밖임을 알 수 있고 6.7g/km 이상이 나올 확률은 1.13%로 회사가 주장하는 6g/km 는 잘못된 값이고 이 값보다 더 높은 값이 100 대의 디젤 버스의 NOx 의 배출량이 됩니다.

4. The Sampling distribution of the Proportion

binomial distribution 을 normal distribution 으로 처리할 수 있도록 될 경우 적용을 할 수 있는 것으로 예를 들어 휴대폰 제조업체가 자신의 휴대폰 구매자가 전체 구매자에 26%라고 발표를 했을 때 별도로 휴대폰 사용자 300 명에게 조사를 했을 때 70 명이 사용한다고 응답을 한다면 23%로 회사가 발표한 26%가 sample error 로 인한 것인지 아님 조사가 잘 못 된 것인지 알아 볼 필요가 있습니다. 이를 위해 sample mean 들의 평균 π로 0.26 를 사용하고 표준 편차는 수식 7.7 에 적용을 하면 됩니다.

$$\rho_p = \frac{\sigma}{\sqrt{n}} = \sqrt{\frac{\pi(1-\pi)}{n}} \tag{7.7}$$

여기서, n 은 총 조사한 휴대폰 사용자 입니다.
그럼 z-score 를 계산하기 위해 수식 7.5 를 적용하면 다음과 같습니다.

$$z_p = \frac{p-\pi}{\rho_p} \tag{7.8}$$

앞의 예를 수식 7.8 에 적용을 하면 다음과 같습니다.

```
> sigma = Math.sqrt( (0.26 * (1-0.26))/300 )
0.025324559884296775

> p = 70/300
0.23333333333333334

// z-score
> (p - 0.26)/sigma
-1.052996252985313

> jMath.stat.normcdf( p, 0.26, sigma )
0.1461713546999156
```

결과를 보시면 0.26 이 평균일 경우 sample 에 비율이 0.23 보다 작을 확률은 14.6%로 확률 관점에서 이 확률이면 26%를 신뢰가 있는 평균값으로 고려할 수 있습니다. 즉 sample error 때문에 발생된 문제라 판단할 수 있습니다.

만일 Population 의 개수(N)가 작아 population 에 sample 의 비율 n/N 이 5%보다 크다면 수식 7.7 은 수식 7.6 과 같이 Correction factor 로 보완이 되어야 합니다.

$$\rho_p = \sqrt{\frac{\pi(1-\pi)}{n}}\sqrt{\frac{N-n}{N-1}} \tag{7.9}$$

예를 들어 회사 직원이 800 명인 회사에서 영어 회화가 가능한 직원의 비율이 51%라고 합니다. 그런데 100 명의 직원을 선별해서 영어 회화를 시켜보니 40 명만이 영어 회화가 가능한 것으로 조사 될 때 회사 직원 전체의 회화 가능 인원은 51%보다 낮게 봐야 되는지 알아 보려고 합니다. 그런데 조사 대상자는 100/800 은 12.5% 비율이므로 표준 편차는 7.9 를 적용해야 합니다.

```
> pi = 0.51
0.51
> p = 0.4
0.4

> sigma = Math.sqrt( (pi * (1-pi))/100 ) * Math.sqrt( (800-100)/799 )
0.04679061553482343

// z-score
> (p - pi)/sigma
-2.350898759135443

> jMath.stat.normcdf( p, pi, sigma )
0.009364064350817491
```

z-score 는 -2.35 로 95%범위인 -1.96 ~ 1.96 범위 밖이고 40%보다 낮을 확률을 보면 0.9%로 매우 낮게 나타납니다. 만일 조사 대상자들이 회사에서 우연히 영어를 가장 못하는 100 명만 선출된것이라면 이러한 결과를 얻을 수 있을지 모르지만 그럴 확률이 매우 낮기 때문에 실제 영어 회화 가능한 인원은 51%보다 적다고 판단할 타당성을 갖고 있습니다.