# A Gentle Introduction to Slightly Advanced Statistical Methods for Behavioral Research

Himanshu Mishra [1]

July, 2015

[1]For more information email: himanshu.mishra@utah.edu

# Contents

# Chapter 1

# Introduction and Software Basics

These notes are prepared for a summer seminar on statistical methods taught at the University of Utah in summer 2015. Methods discussed in this seminar are essential for experimental work but are not readily accessible. Many times such methods are taught across various statistics courses and prevent researchers from seeing how they can be easily used in their research papers.

This seminar assumes that you have taken some basic experiment data analysis course and are familiar with t-test, ANOVA etc.

The topics we will cover are power analysis, outlier detection, effect size, bootstrap effect size, quantile regression, fallacies of NHST, regression discontinuity designs, bayesian t-test and bayesian anova. These notes currently don't cover details of regression discontinuity designs, bayesian t-test and bayesian anova.

Each of the topics is accompanied with an software code. We will be using R in this seminar so please install R from `https://cran.r-project.org/`. If you are new to R read "A (very) short introduction to R" `http://cran.r-project.org/doc/contrib/Torfs+Brauer-Short-R-Intro.pdf`

To make your R experience simpler, feel free to use R Studio (`https://www.rstudio.com/products/RStudio/`)

In only a few occasions (e.g., Monte Carlo Power Analysis) we will be using Python. The best place to install Python is Anaconda from `http://continuum.io/downloads`. If you want to learn more about Python, here is a good introductory book `http://www.kevinsheppard.com/images/0/09/Python_introduction.pdf`

A caveat: Most of the material in these notes is compiled from various sources. All efforts have been made to cite appropriate sources. Any omission to cite a source is not deliberate.

# Chapter 2

# Power Analysis

What is Power Analysis ? Let's first understand some very simple concepts that we will be using throughout the course. It is always easier to understand them through an example. Imagine we ran a study to find out if 20% price discount is more preferred than 20% bonus quantity. We chose soap as our stimulus and recruited 60 participants. 30 were randomly assigned to the price discount condition (P condition) and 30 to quantity discount condition (Q condition). Our dependent variable was people's willingness to buy soap on a 1-7 scale. We found out that in the P condition the mean willingness to buy was 5.9 and in the Q condition it was 4.9, $p < .05$. Therefore, in this sample we find that price discounts are more preferred than quantity discounts. Now the question is does this result mean that consumers as a whole would prefer price discounts to quantity discounts? In other words, does our sample (of 60 participants) provide us any reliable information about the population (all consumers who are exposed to price/quantity discounts)?

Samples are subsets of the population. While samples are drawn from the population, what we see in samples may not reflect what is happening in the population. Why? Samples don't always capture the true nature of the population because of Sampling error . Whenever we draw a subset of population we run the risk of introducing sampling error. One way to think about sampling error is to understand it as uncertainty in our conclusion from the sample. Therefore, two possible conclusions about our study can be drawn:

> **Conclusion 1:** The mean willingness to buy soap with price or quantity discount are ***actually the same in the population***. Sampling error is responsible for the difference observed in the study.
>
> **Conclusion 2:** The mean willingness to buy soap with price discount is ***actually large compared to the quantity discount in the population***. Sampling error is ***not*** responsible for the difference observed in the study.

Conclusion 1 is commonly referred to as the null hypothesis[1] and Conclusion 2 as the alternate hypothesis .

Now let's consider two possible realities at the population level. These realities exist regardless of what our sample says.

[1]The null hypothesis can be stated in many different ways. For instance, we can have a null stating that the difference between the P and Q conditions is .5.

**Reality1:** The willingness to buy soap with ***price or quantity discount are not different***.

**Reality 2:** The mean willingness to buy soap with price discount is ***actually very large*** .

So what happens when our conclusions don't match with the reality? Simple answer: Type I and Type II errors ☺. Let's understand why.

| | *Reality 1 (no difference)* | *Reality 2 (yes difference)* |
|---|---|---|
| *Conclusion 1(no difference)* | ☺ $p = 1 - \alpha$ | ☹ Type II error, $p = \beta$ |
| *Conclusion 2 (yes difference)* | ☹ Type I error, $p = \alpha$ | ☺ *power* $= 1 - \beta$ |

This table says the following: if reality 1 matches with conclusion 1 then we are happy ☺. This is the reason we set $\alpha > .95$ and thus $p < .05$. In other words, we are trying to avoid Type I error . So when we say $p < .05$ we are implying that the probability of incurring Type I error is less than .05. One way to understand this is to say that we want to minimize the chance of concluding that price discount is more preferred than quantity discount (conclusion 2) when in reality there is no difference between them (reality 1).

Type II error is drawing conclusion 1 (no difference) when a difference actually exists (reality 2). Just like $\alpha$ plays an important role in assessing the chance of committing a Type I error, $\beta$ plays an important role in assessing the chance of committing a Type II error. $\beta$ tells us what are the chances that we would conclude that no difference exists between price discount and quantity discount conditions when in reality there is a difference between them.

Power is simply $1 - \beta$. As you can guess, we ideally want to keep $\beta$ low so that the power of our study is high. A more formal way to define the power of a study is the following: *It is the probability of rejecting the null hypothesis (conclusion 2) when the null is really false (reality 2).*

## 2.1 Ingredients of Power

What influences the power of a study? Actually, it is quite straightforward.

1. Sample Size: Higher sample sizes increase the power of a study.
2. Effect Size: Essentially effect size is [standardized difference/magnitude of a phenomenon] (see more details in the note on Effect Size). Let's consider our study. If we assume that standard deviations of P and Q conditions are 1, then cohen's d (one of the commonly used effect size measure) for our study would be 1 [2]. The important thing to remember about effect sizes is that they don't change with changes in the sample size.

[2] Cohen's d = $\frac{Mean_1 - Mean_2}{s}$ where $s$ is the weighted average of each group's standard deviation.

3. $p$ value: if we keep Sample size and Effect size constant, a study will have less power if $p < .01$ than if $p < .05$. In other words, your study will have higher power if you lower the standards to reject the null hypothesis (i.e., increase the $p$ value).

From a practical standpoint, if a researcher wants to increase the power of a study usually his best bet is increasing the sample size. Why? Because $p$ values are implicitly agreed upon in research areas (like $p < .05$) and effect sizes depend a lot on the research questions one is trying to answer. Thus, it is easier to change the sample size than $p$ value and effect size.

## 2.2 The use of Power

The most common use of power analysis is in calculating the sample size of a proposed study. How could you do that? Without going into many technical details, we know that power depends on sample size, effect size and $p$ value. If we provide values of effect size, $\alpha$ value (i.e., $1 - p$), and the power we want our study to have (i.e., $1 - \beta$), we can calculate the sample size we need to have in our proposed study.

You must be wondering that if I know only $\alpha$ (normally it is .05), how can I provide values of effect size and $\beta$? With effect size it almost seems like a catch 22 situation. I need effect size to calculate the sample size of a proposed study, but how can I know the effect size if I have not even run the study? There are 3 ways to address this problem: a) run a pilot study and calculate the effect size, b) if you have already run a study, use its effect size, c) if the effect you are testing is based on some existing theory, see effect sizes of existing studies related to that theory. Use the average effect size observed as your input.

Unlike the value of $\alpha$ there is no clear guideline for the value of $\beta$. However, you can use some thumb rules. Less than .50 power is a bad idea. .90 and above is really good but to achieve that level of power, your sample size needs to be very high (sometimes it is practically impossible to collect data from such large samples). An arbitrary yet practical advice is to keep power around .80.

> Remember, your sample size calculation based on power analysis are as good as the value you use for the effect size. So in reporting, be fully transparent about how you chose the effect size value.

> Considering all the assumptions that need to be made, sample size calculations are essentially hypothetical.

## 2.3 How can I calculate Power?

If you want a one-stop-shop solution for most (not all) of your power analysis needs, you can use the stand-alone program G*Power (http://www.gpower.hhu.de/)

R has some useful packages that will perform power analysis for standard designs (see R package 'pwr' http://cran.r-project.org/web/packages/pwr/index.html). Here is a

simple example. If the data from our discount study is $Mean_P = 5.9, SD_p = 1, Mean_Q = 4.9, SD_Q = 1$ then the effect size will be 1 (as we discussed earlier). The R code for the power analysis with the proposed sample size of 40 will be

```
require(pwr)
pwr.t.test(d=1,n=40,sig.level=0.05,alternative="greater") # here d is the
    effect size, n is the proposed sample size and sig.level is alpha
```

The output will show that with the proposed sample size of 40, our study's power will be .99.

While very convenient, there is one problem with canned power analysis programs. If your design is complex you may not find a good answer. Why? Because for complex designs, many times there are no analytical solutions available to estimate power. In such situations a solution can be found in Monte Carlo simulations. Considering this course is about advanced experiment analysis techniques, let's understand how you can estimate power for nearly any design with simulation.

## 2.4 Monte Carlo Power Analysis

First, let's quickly understand what is Monte Carlo Analysis. For many problems it is hard to find a closed-form solution (i.e., to have a formula that you can use to find the unknown quantity). Monte Carlo analysis is used to solve such problems. The basic premise is the following: if we repeatedly sample from a known probability distribution/process, we can numerically obtain approximate solutions. Let's take the example of the Central Limit Theorem (CLT) to understand Monte Carlo analysis. Here we have a mathematically derived prediction to compare the simulation results.

The CLT says that if we take several random samples of size $n$ from any distribution with true mean $\mu$ and standard deviation $\sigma$, the distribution of the sample means will follow a normal distribution with mean $\mu$ and standard deviation $\frac{\sigma}{\sqrt{n}}$. Here is an $R$ code.

```
X = matrix(rnorm(10000 * 15, 5, 3), 15) # Here we are generating 10,000
    samples # with 15 observations in each sample. The population has mean 5
    and sd 3
drawmean = apply(X, 2, mean) # taking mean of each of the sample. We get 10000
    # means.
hist(drawmean) # plot them
mean (drawmean) # check and see if the mean of means matches with the
    population mean 5.
```

Now, you can play with sample size. Change it from 15 to 30 to 100 and see what happens. You will notice that as you increase the sample size, the standard deviation of the distribution of the sample means will decreases (just like what CLT predicts). Later we will discuss how CLT helps us in forming confidence intervals around the mean.

Now let's revisit the price vs. quantity discount study we discussed in the beginning of this section. This is the simplest example to understand how simulation can be used to estimate power for different sample sizes.

Again, here is the data from our discount study $Mean_P = 5.9, SD_p = 1, Mean_Q =$

4.9, $SD_Q$ = 1 If you recall from Type 1 and Type 2 error table, Power is 1 – $\beta$. That is the probability of rejecting the null when it is false (i.e., conclusion 2, reality 2). Therefore, given values of means and SDs of our discount study, we can sample many times and calculate the proportion of times we find that the P and Q conditions are different at $p < .05$ significance level. This proportion is essentially Power. Such calculations of power has become possible due to the availability of abundant computing power. So here are the steps to calculate power. Assume we want to find out what would be the power if we repeat this study with 80 participants in each condition.

1. Draw a sample of 80 observations from a normal distribution with mean = 5.9 and SD =1 (i.e., P condition)
2. Draw a sample 80 observations from a normal distribution with mean = 4.9 and SD =1 (i.e., Q condition)
3. Run a T_test and if you observe $p < .05$, then count it as 1 else count it as 0
4. Repeat steps 1-3, 1000 times.

The proportion of times you find $p < .05$ in 1000 iterations, is the Power estimate for sample size of 80. You can repeat this process for any sample size. The following is a python code that calculates power for sample sizes between 20 and 100:

```
%matplotlib inline


import numpy as np
import pylab as plt
from math import sqrt
from scipy import stats
import random
from decimal import Decimal
from __future__ import division
import time



mean1= 5.9 # condition 1 mean
mean2= 4.9 # condition 2 mean
sd1 =1 # condition 1 SD
sd2=1# condition 2 SD
alpha=.05 # significance level
iter = 500 # number of samples to interate
rep=[] # ignore these are placeholders
sample_n=[]# placeholders

# this calculates Cohen's d
def cohen_d(mean1,mean2,sd1,sd2):
       return ((mean1 - mean2) / (sqrt((sd1 ** 2 + sd2 ** 2) / 2.0)))


for i in range (20,100,10):# sample size range with increment of 10
    #print('Sample Size:',i)
    millis = int(round(time.time() * 1000))
    random.seed(millis*i)# to ensure random draws
```

```
    count = 0;
    n=i;

    for g in xrange(1,iter):
        random.seed(int(round(time.time() * 1000))*g)
        rand1=random.randint(445, 200000)
        np.random.seed(seed=rand1*g)
   # drawing samples from condition P
        y1 = np.random.normal(mean1,sd1, size=n)
   # drawing samples from condition Q
        y2 = np.random.normal(mean2,sd2, size=n)

        y = np.concatenate((y1, y2))
        t_stat1= stats.ttest_ind(y1,y2,equal_var = False)
        b = np.array(t_stat1)
        tval,p_val = np.hsplit(b,2)
        if p_val < alpha:
            count = count+1

    f1=count/iter
    print("Sample Size:%s, Power:%s)" %(i,f1))
    rep.append(f1)
    sample_n.append(i)
plt.title("Cohen\'s d: %s" %(cohen_d(mean1,mean2,sd1,sd2)))
plt.ylabel("Power",fontsize=12,fontweight='bold')
plt.xlabel("Sample Size per condition",fontsize=12,fontweight='bold')
plt.scatter(sample_n,rep,color='r')
plt.plot(sample_n,rep)
plt.show()
```

The output shows:

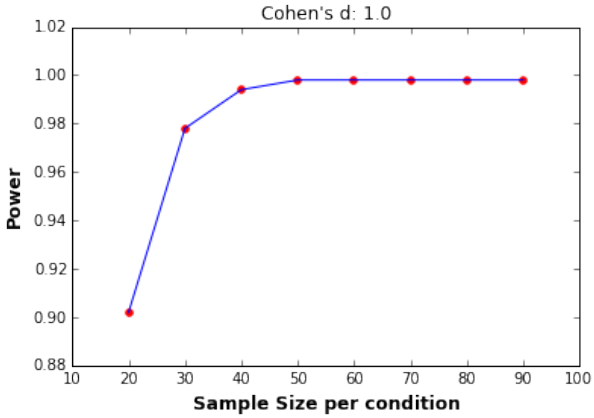


Figure 2.1: Power Simulation for reduced effect size cell design

Earlier we discussed what influences the power of a study. One factor was the sample size. As you can see in this figure as sample size increases, power increases. Simulations also help us understand how effect size changes power. Let's imagine that the SDs of our discount study are 2 instead of 1 .i.e., $Mean_P = 5.9, SD_p = 2, Mean_Q = 4.9, SD_Q = 2$. This changes our effect size (Cohen's d) from 1 to .5. The following is a figure of simulated power with this information

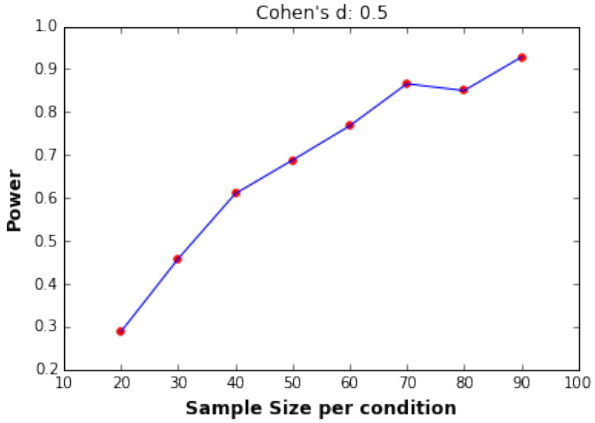


Figure 2.2: Power Simulation for 2 cell design

As you can see that as effect size decreases you need a higher sample size to achieve the same power that you achieved with a smaller sample size and high effect size. Compare the power in figure 2.1 and figure 2.2 for the same sample sizes. As an example, for a sample size of 20 when Cohen's d changes from 1 to .5, the power of your future study will drop from .9 to .3.

### 2.4.1 2x2 ANOVA Power Simulation

One of the most commonly used experiment design is the 2x2 between-participants design where data is usually analyzed by ANOVA. Here is a power simulation code for such designs

```
## ANOVA-Monte Carlo Power Simulation
%matplotlib inline

import pandas as pd
import numpy as np
import pylab as plt
from math import sqrt
from scipy import stats
import random
from decimal import Decimal
from __future__ import division
import time
from statsmodels.formula.api import ols
from statsmodels.stats.anova import anova_lm

mean_A1B1= 5.9
sd_A1B1 =2.2

mean_A1B2= 5.1
sd_A1B2 =1.8

mean_A2B1= 4.9
sd_A2B1 =1.2

mean_A2B2= 5.4
```

```
sd_A2B2 =1.8

alpha=.05 # significance level
# number of samples to interate
iter = 1000
# ignore these are placeholders
rep_a=[]
rep_b=[]
rep_ab=[]
sample_n=[]# placeholders

for i in range (5,100,10):# sample size range with increment of 10
    #print('Sample Size:',i)
    millis = int(round(time.time() * 1000))
    random.seed(millis*i)# to ensure random draws
    count_a = 0
    count_b = 0
    count_ab =0
    n=i;
    x1 = np.zeros([n,1]); x2 = np.ones([n,1]);
    x3 = np.ones([n,1]); x4 = np.zeros([n,1]);

    X1 = np.hstack((x1, x2));
    X2 = np.hstack((x1,x4));
    X_1= np.concatenate((X1, X2));

    X3 = np.hstack((x3, x2));
    X4 = np.hstack((x3, x4));

    X_2 = np.concatenate((X3,X4))

    factor= np.concatenate((X_1,X_2))
    for g in xrange(1,iter):
        random.seed(int(round(time.time() * 1000))*g)
        rand1=random.randint(445, 200000)
        np.random.seed(seed=rand1*g)


        y1 = np.random.normal(mean_A1B1,sd_A1B1, size=n)
        y2 = np.random.normal(mean_A1B2,sd_A1B2, size=n)
        y3 = np.random.normal(mean_A2B1,sd_A2B1, size=n)
        y4 = np.random.normal(mean_A2B2,sd_A2B2, size=n)
        depv= np.concatenate((y1,y2,y3,y4))
        depv=depv[:,None]

        data = np.hstack((depv,factor))

        df = pd.DataFrame(data, columns=['dv', 'A', 'B'])
        formula = 'dv ~ C(A) + C(B) + C(A):C(B)'
        lm = ols(formula, df).fit()
        #print anova_lm(lm)
        jj= np.array(anova_lm(lm))
        a_pval= jj[0,4]
        if a_pval < alpha:
            count_a = count_a+1
        b_pval= jj[1,4]
```

```
        if b_pval < alpha:
            count_b = count_b+1
        ab_pval= jj[2,4]
        if ab_pval < alpha:
            count_ab = count_ab+1
    fa=count_a/iter
    print("Sample Size:%s, Power for main effect 'A':%s)" %(i,fa))
    fb=count_b/iter
    print("Sample Size:%s, Power for main effect 'B':%s)" %(i,fb))
    fab=count_ab/iter
    print("Sample Size:%s, Power for main effect 'A*B':%s)" %(i,fab))

    rep_a.append(fa)
    rep_b.append(fb)
    rep_ab.append(fab)
    sample_n.append(i)
#plt.title("Cohen\'s d: %s" %(cohen_d(mean1,mean2,sd1,sd2)))
plt.ylabel("Power",fontsize=12,fontweight='bold')
plt.xlabel("Sample Size per condition",fontsize=12,fontweight='bold')
plt.scatter(sample_n,rep_a,color='r',label='A')
plt.plot(sample_n,rep_a,color='r')

plt.scatter(sample_n,rep_b,color='k',label='B')
plt.plot(sample_n,rep_b,color='k')

plt.scatter(sample_n,rep_ab,color='g',label='A*B')
plt.plot(sample_n,rep_ab,color='g')
plt.legend(loc=4)
plt.show()
```

## 2.5 The fallacy of post-hoc Power Analysis

Until now we discussed how power analysis uses an existing study (or studies) to help us estimate the sample size for a future study. There is another (often contentious) issue of using power analysis to interpret an existing study's results. Post-hoc power is essentially estimating the (post-hoc) power of an already concluded study. Going back to our discount study example, this would mean estimating what was the power of that study. Why is such a power analysis contentious? The issue becomes contentious when such a power analysis is used to interpret non-significant study results.

Let's assume in our discount study, our results showed that $p = .4$ i.e., we were unable to reject the null hypothesis indicating that there is no difference between price and quantity discount conditions. Now if we use power analysis to understand why we were unable to reject the null hypothesis, we are committing a logical fallacy. Why? Because in these situations, power is an inverse function of observed $p$ values (see Hoenig and Heisey (2001) for further detail). So we gain no new knowledge by calculating power of statistically non-significant results. Bottom line, use power analysis to estimate sample size of future studies.

### Further Readings

If you want to explore this topic further, there are countless excellent articles and books. Here are some (not an exhaustive list): Maxwell et al. (2008), Liu (2013).

# Chapter 3

# Outliers

"An outlier is an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism" - Hawkins (1980)

Outlier detection has a long history. Empirical observations, especially responses collected from humans, invariably contain outliers. Most of the commonly used methods made assumptions about the underlying mechanism that generates the data. They assumed data to be normally distributed and focused on univariate data. There exists more than 100 such methods. One way to understand these methods is divide them into categories of robust vs. non-robust. Let's start with non-robust methods.

One of the most commonly used method in social sciences (particularly in psychology and consumer research) is removing univariate observations that are more than 3 standard deviations away from the mean (the $3SD$ rule). What is the rationale behind this method? We know from the properties of normal distribution that the probability of points lying beyond $3SD$ limit on either side of the mean is less than 0.5% as the figure on the next page shows [1].

Why is this $3SD$ rule non-robust and what is the problem with this method? The value of the mean ($M$) and standard deviation ($SD$) used to detect outliers are affected by the very same outliers that one is trying to remove. Let's take a simple example. Assume we collected data from 6 respondents about their happiness on a 1-100 scale. The following are their ratings:

{20, 25, 30, 12, 15, 95}

This data has $M$ = 32 and $SD$ = 31.14. Now while we can see that 95 is an outlier, we can't remove it using the $3SD$ rule because the presence of 95 has changed the $M$ and $SD$.

## 3.1 So what are better alternates to using the $3SD$ rule?

There are many robust alternates to using the $3SD$ rule. Here we discuss two of them.

[1] Credit:Wikipedia

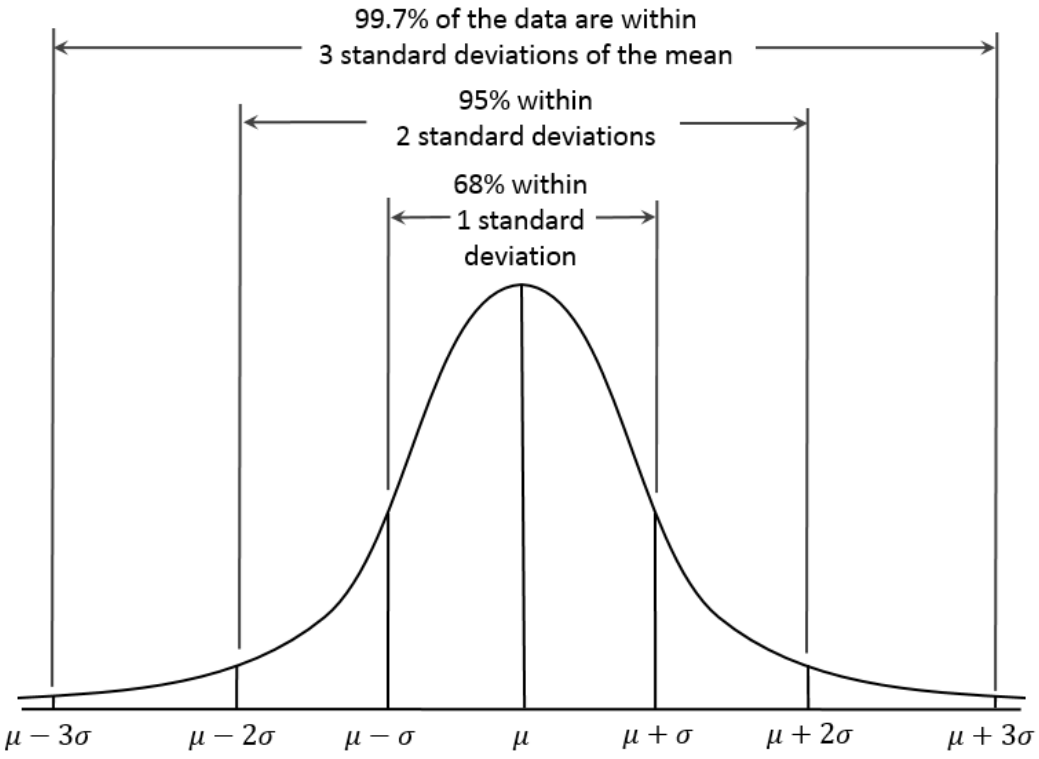


Figure 3.1: Normal Distribution

### 3.1.1 Median Absolute Deviation

The Median is very insensitive to the presence of outliers which means adding outliers does not change the median. In general, insensitivity of any estimator (mean, median etc.) to outliers is measured by their breakdown point. So first let's understand what is the breakdown point.

**Breakdown point:** The estimator's breakdown point is the maximum percentage of observations that can be completely contaminated (i.e., equal to infinity) before making the estimator to result in a false value. For example, when a single observation has an infinite value, the mean of all observations becomes infinity. This shows that the breakdown point of mean is 0. The median value on the other hand has breakdown point of .5, which means we can have up to 50% observations that are infinity ($\infty$) before median goes completely haywire. This makes median preferable to mean. The following example highlights this

For dataset $[2, 2, 5, 8, 8, 9, 9, 10, 22, 28, 36]$ the median is 9.

Now if we multiply the last 5 values of this dataset by 1000. The dataset will be $[2, 2, 5, 8, 8, 9, 9000, 10000, 22000, 28000, 36000]$ but the median will again be 9.

This property of the median makes outlier detection based on Median Absolute Deviation (**MAD**) a very good alternate to the $3SD$ rule.

First let's understand what is MAD. It is the median of absolute deviations from the median. Formally,

$$MAD = median(|x_i - median_j(x_j)|)$$

Let's take an example to understand this. If our data is [2, 2, 3, 5, 2, 7], then the median (i.e., $median_j(x_j)$) of this data is 2.5.

$|x_i - 2.5|$ for each observation will be $[|2-2.5|, |2-2.5|, |3-2.5|, |5-2.5|, |2-2.5|, |7-2.5|] = [0.5, 0.5, 0.5, 2.5, 0.5, 4.5]$. The median of these values is MAD. So for our dataset MAD = 0.5.

Similar to the median, MAD also has a breakdown point of 0.5 which makes it better than sample standard deviation in the presence of outliers. If the underlying distribution in the absence of any outlier is assumed to be normal then MAD is multiplied by 1.4826. The rule for detecting outliers is:

$$X > Median + 3 * 1.4826 * MAD$$

Or

$$X < Median - 3 * 1.4826 * MAD$$

The following code removes outliers using the MAD

```
from numpy import median, absolute

def mad(data, axis=None):
    return 1.4826*median(absolute(data - median(data, axis)), axis)

data=np.array([-30,1, 4, 8, 8, 3, 4, 5,9, 90])

mad = mad(data)
ul = median(data)+3*1.4826*mad
ll = median(data)-3*1.4826*mad
idx= np.where((data > ul)|(data<ll))

print ('median absolte deviation', mad)

print ('outlier locations:', np.ndarray.flatten(np.array([idx])+1))

print ('outliers', data[(data > ul)|(data<ll)])

data_clean = data[(data < ul)&(data>ll)]

print ('clean data',data_clean)
```

```
## Output
>> ('median absolte deviation', 5.1890999999999998)
>> ('outlier locations:', array([ 1, 10]))
>> ('outliers', array([-30, 90]))
>> ('clean data', array([1, 4, 8, 8, 3, 4, 5, 9]))
```

## 3.2 Boxplot

Tukey proposed boxplots in 1960's. Boxplots are non-parametric, which is a fancy way of saying they don't make any assumptions about the underlying distribution that generated the data (it could be normal, gamma, cauchy, it doesn't matter). Let's create a boxplot to understand what it is

```
from pylab import *
data =[2,2,8,9,9,10,14,28]
boxplot(data,0,'rD') # here 'rD' changes the color and shape of outliers
show()
```

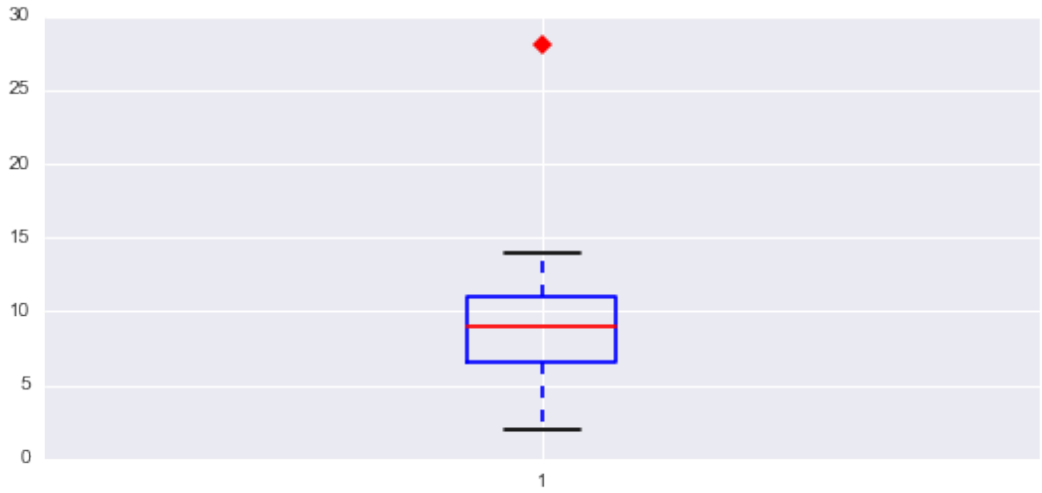


Figure 3.2: Boxplot

There are different parts of this boxplot. Let's start with the redline. It is the median ($50^{th}$ percentile) of the data. The box starts at the first quartile ($25^{th}$ percentile of data) and ends at the third quartile ($75^{th}$ percentile of data). The difference between the first and the third quartile is also called the interquartile range or **IQR**. So within the box lies the middle 50% of the data. In our sample dataset these values are:

Median: 9.0

$1^{st}$ quartile: 6.5

$3^{rd}$ quartile: 11

IQR: 11 - 6.5 = 4.5

The lower black whisker shows the point within 1.5 x IQR below the $1^{st}$ quartile. In our data this will be the 6.5 ($1^{st}$ quartile) - 1.5 x 4.5 (IQR) = -.25. So the smallest data point between 6.5 and -.25 would be 2.

The upper black whisker shows the point within 1.5 x IQR above the $3^{rd}$ quartile. In our data this will be 11 ($3^{rd}$ quartile) + 1.5 x 4.5 (IQR) = 17.75. So the largest data point between 11 and 17.75 would be 14.

The red diamond shows the outlier in our data.

### 3.2.1 How can one determine outliers in boxplot?

There are 2 rules. One for clear outliers and the other for suspected outliers.

**Clear Outliers:** Data points that fall 3 x IQR or more above the third quartile or 3 x IQR or more below the first quartile. If we look at out data, this limit is 3 x 4.5 = 13.5. The third quartile is 11 so any point above 13.5 + 11 = 24.5 would be a clear outlier. If you see our dataset, 28 would qualify as a clear outlier.

**Potential Outliers:** They are 1.5 x IQR or more above the third quartile or 1.5 x IQR or more below the first quartile. In our data set, 1.5 x IQR value above the third quartile would be 11 + (1.5 x 4.5) = 17.75 and 1.5 x IQR value below the first quartile would be 6.5 - (1.5 x 4.5) = -.25.

**Would the $3SD$ rule work?** Our dataset also highlights why many times the $3SD$ rule is not a good choice to remove outliers. The sample mean of our data is 10.25 and the sample standard deviation is 7.69. According to the $3SD$ rule we can consider a data point an outlier only if it is more than 10.25 + 3 x 7.69 = 33.32 or less than 10.25 - 3 x 7.69 = -13.12. Since our outlier 28 has influenced the sample mean and SD, we can not label 28 as an outlier.

**Modified Boxplot Rule:** Carling (2000) proposed a modification to the boxplot rule. This modification was motivated by the fact that sample size influences which observations are considered outliers. According to this modification an observation $X$ is an outlier if

$$X > Median + k * IQR$$

Or

$$X < Median - k * IQR$$

Where if $N$ is the sample size then

$$k = \frac{17.63N - 23.64}{7.74N - 3.71}$$

Here is a code that removes outliers using Carling's modification

```
%matplotlib inline
from pylab import *
import numpy as np
import seaborn as sns

data =np.array([-30,-24,2,2,5,8,8,9,9,10,22,36,44])

iqr = np.percentile(data,75)-np.percentile(data,25)
median = np.median(data)
k= (17.63*len(data)- 23.64)/(7.74*len(data)-3.71)
ul = median+(k*iqr)
ll = median-(k*iqr)
```

```
idx= np.where((data > ul)|(data<ll))

print ("outlier locations:", np.ndarray.flatten(np.array([idx])+1))
print ("outliers", data[(data > ul)|(data<ll)])
data_clean = data[(data < ul)&(data>ll)]
print ("clean data",data_clean)
```

here is the output.

```
>> (('outlier locations:', array([ 1, 2, 12, 13]))
>> ('outliers', array([-30, -24, 36, 44]))
>> ('clean data', array([ 2, 2, 5, 8, 8, 9, 9, 10]))
```

## 3.3 Outliers in Multivariate data

Up till this point, our discussion of outliers has been confined to finding outliers in univariate data. However, many times the data is multivariate (e.g., repeated measures, multiple dependent variables, covariates etc.) so next we will see how we can detect outliers in a multivariate dataset.

The most common question that appears in the context of multivariate outliers is why can't we just use univariate methods on each of the variables and detect outliers? Why do we need a separate set of methods? Let's take an example to understand why. Imagine we have data on two variables, age and systolic blood pressure. If these data points are coming from people in the age group of 16 to 75, a systolic blood pressure of 145 may not appear to be an outlier on its own as this number increases with age and in our sample we may find many people having this or a higher number. So we may not be able to identify a 155 reading as an outlier if we search for outliers using univariate methods. However, if we consider age and systolic blood pressure together, we realize that this datapoint is associated with someone who is 16 years old. While a 155 systolic blood pressure is not uncommon among 45-55 year olds, it is certainly an outlier among 16 year olds. As this example highlights we need methods that uses all the relevant variables to detect outliers.

How can we find outliers in multivariate settings? As you would imagine, a large body of literature exists on this topic. Historically, the major shift in this areas has been the use of robust methods to detect outliers in such datasets. So we will discuss mostly robust methods but to give you an intuitive understanding of how outliers are detected in multivariate datasets, we will use simpler non-robust methods. Let's take the non-robust example of univariate data. As we discussed earlier, standard deviation is used as a proxy for distance: high SD+$\mu$ $\rightarrow$ high distance thus, an outlier. In multivariate data, Mahalanobis distance can be used to identify an outlier. First, let's take a simple example to learn how to calculate Mahalanobis distance. Continuing our example of age and systolic blood pressure, if our data is as follows

| Person # | Age | Systolic Blood Pressure |
|---|---|---|
| 1 | 16 | 155 |
| 2 | 20 | 120 |
| 3 | 18 | 125 |
| 4 | 45 | 160 |
| 5 | 50 | 150 |
| 6 | 60 | 160 |

Just by looking at the data, person 1 appears to be an outlier. However, our goal is to find an outlier using some form of distance so we can extend this method to data with hundreds of observations and with more than 2 variables. One option we have is to calculate the Euclidean distance of each person from the center (centroid) of this sample. So what is the center of this sample? It is simply the average of age and the average of systolic blood pressure i.e., 34.83 and 145.00. So if we want to measure the distance of person 2 (P2) from this center, we will use the formula $d_e(P2, center) = \sqrt{(20 - 34.83)^2 + (120 - 145)^2} = 29.06$.

| Person # | Euclidean Distance ($d_e$) from the center |
|---|---|
| 1 | 21.32 |
| 2 | 29.06 |
| 3 | 26.13 |
| 4 | 18.12 |
| 5 | 15.97 |
| 6 | 29.30 |

Clearly, this method does not pick the right outlier (person 1). He doesn't appear that far from the center. Why is the simple Euclidean distance not a good way to measure distance? First, the Euclidean distance is very sensitive to the scales of the variables (age and systolic BP). In standard geometric measurements, all variables are measured in the same units of length (meter, inches etc.). But with the kind of data we use it is rarely the case. Our variables measure constructs where scales are not comparable (e.g., age, emotional reaction, amount spent etc.).

Second, the Euclidean distance completely ignores correlated variables. In our hypothetical case, the Euclidean distance has no way of understanding correlation between age and systolic BP.

### 3.3.1 Mahalanobis Distance based methods

The Mahalanobis Distance addresses some shortcoming of Euclidean distance. The Mahalanobis distance takes into account the covariance among the variables in calculating distances. With the Mahalanobis distance the problems of scale and correlation inherent in the Euclidean distance are no longer an issue. Here is a geometric explanation: when using Euclidean distance, the set of points equidistant from a given location (e.g., the centroid) form a sphere. The Mahalanobis distance stretches this sphere to correct for the respective scales of the different variables and to account for correlation among

variables [2].

The formula for calculating the Mahalanobis distance shares a lot of similarity with the Euclidean distance formula. The Euclidean distance can also be written in the form of a matrix dot product as $d_e(x,y) = \sqrt{(x-y)^T(x-y)}$. The Mahalanobis distance is $d_M(x,y) = \sqrt{(x-y)^T S^{-1}(x-y)}$. Here $S^{-1}$ is the inverse of the variance/covariance matrix (in some books it is denoted as $\sum^{-1}$).

Let's use this formula to calculate distance in our age systolic BP example. First, we need to estimate the variance/covariance matrix of our data. It is

| | Age | Systolic Blood Pressure |
|---|---|---|
| age | 364.97 | 225 |
| sbp | 225 | 320 |

Here the diagonal values are variance and off diagonal covariance. Using basic matrix algebra it's inverse is

$$S^{-1} = \begin{pmatrix} 0.0048 & -0.003 \\ -0.003 & 0.0055 \end{pmatrix}$$

if $c$ is the center with values 34.83 and 145 (as we calculated earlier) then $(p1 - c) = \begin{pmatrix} -18.83 & 10.0 \end{pmatrix}$. Here $p1$ is the first person. It is quite straightforward to calculate the Mahalanobis distance from these values. For person 1, the Mahalanobis distance from the center is

$$d_m(p1,c) = \sqrt{\begin{pmatrix} -18.83 & 10.0 \end{pmatrix} \begin{pmatrix} 0.0048 & -0.003 \\ -0.003 & 0.0055 \end{pmatrix} \begin{pmatrix} -18.83 \\ 10.0 \end{pmatrix}} = 1.88$$

Similarly, we can calculate the distance of other people from the center. The following table compiles these values

| Person # | Mahalanobis Distance ($d_m$) from the center |
|---|---|
| 1 | 1.88 |
| 2 | 1.41 |
| 3 | 1.13 |
| 4 | 0.83 |
| 5 | .85 |
| 6 | 1.31 |

As this table shows, person 1 is most distant from the center. Is he an outlier? In a large sample (ours was very small with just 6 values), Mahalanobis squared distances (i.e., $d_m^2$) follow a Chi squared distribution. This makes our life much easier, just like in the univariate case we can set the threshold to $p < .001$ or any other value, find the critical value from the chi square table and classify the points that have squared Mahalanobis distance more than this critical value. So far everything looks good. But as we discussed with univariate outlier detection, such non-robust methods are not optimal for multivariate outlier detection as well. Why? This method breaks down if instead of lone wolf outliers you have a cluster of outliers. The estimation of mean and covariance is influenced by the cluster of outlying points which results in the distance

[2] see this post for a detailed explnation : `https://chrisjmccormick.wordpress.com/2014/07/21/mahalanobis-distance/`

of the outlying point from the mean appearing small (also known as masking). So what's the right method?

### 3.3.2 Robust outlier detection methods

There are many robust alternates. We will focus on Minimum Covariance Determinant (MCD) proposed by Rousseeuw (1984)(the fast-MCD algorithm that implements it was developed by Rousseeuw and Driessen (1999)).

If your data has $n$ observations and $p$ variables and assuming $n > 2p$. The first step is the user input of a number $h$ , which should be $\frac{n+p+1}{2} < h < n$.

1. Then the algorithm selects $h$-sized subsamples of $n$. Intuitively, the value $h$ can be perceived as the minimum number of points that must not be outliers (e.g., if $n$ = 50 and you think at least 60% of these points are not outliers then $h$ can take any value between 30 and 49. If you chose $h$ = 35 then subsamples of size 35 are selected).
2. For each subsample, the determinant of its covariance matrix is calculated.
3. The subsample with the smallest determinant is chosen. Intuitively, it is the tightest cluster of $h$ points.
4. The mean and covariance matrix of this optimal subsample is used to calculate the Mahalanobis distance of every point.
5. The last step is the familiar part of comparing each point's Mahalanobis distance with the critical values of the chi square distribution and labeling those points as outliers whose Mahalanobis distance is more than the critical values.

This approach is less susceptible to masking effects. However, it is quite easy to see why finding MCD via this algorithm is very hard. If $n$ = 50 and $h$ = 35 then it requires calculating the determinant of potential $\binom{50}{35}$ = 2250829575120 combinations. The solution is Fast MCD, which is implemented in most statistical programs (if you are interested in knowing how it works read Rousseeuw and Driessen (1999))

### 3.3.3 Working with data

R provides some of the best packages to identify outliers in multivariate data. We will use 2 such packages *robustbase* and *chemometrics*. Let's use our age and systolic BP data to see how you can find outliers.

```
require(robustbase)
require(chemometrics)
Age=c(16,20,18,45,50,60)
Systolic_BP=c(155,120,125,160,150,160)
vect=cbind(Age,Systolic_BP) # combine age and Systolic BP columns
x.mcd=covMcd(vect, alpha=.5) # calculate the Minimum Covariance Determinant
    (MCD) #estimator via the Fast MCD. Here alpha determines the values of h.
    # roughly h= sample size*alpha
x.mcd$mah # prints robust mahalanobis distance
```

```
x.mcd$mcd.wt # prints ‘‘outlyingness’’ of observations. here 0 means outlier
    #and 1 inlier
drawMahal(vect,center=x.mcd$center,covariance=x.mcd$cov,quantile=0.975,pch=13)
    # draws robust 97.5

sum(x.mcd$mcd.wt == 0) #counts number of outliers
sum(x.mcd$mcd.wt == 1) # counts number of inliers
new<-data.frame(x.mcd$mcd.wt,vect) # creates a new data set where 0, 1 outlier
    scores are in the first column
datNew <- new[-which(x.mcd$mcd.wt == 0), ] # this create a new dataset which
    excludes the outliers i.e. it is your clean data
```

Here is the output:

```
> 10.7984320 0.6151340 0.5078648 1.0925750 0.2893012 0.6335340 # Mahalanobis
    distance
> 0 1 1 1 1 1 # Outlier weight, first one is an outlier
```

and the next figure captures the 97.5% tolerance ellipse.

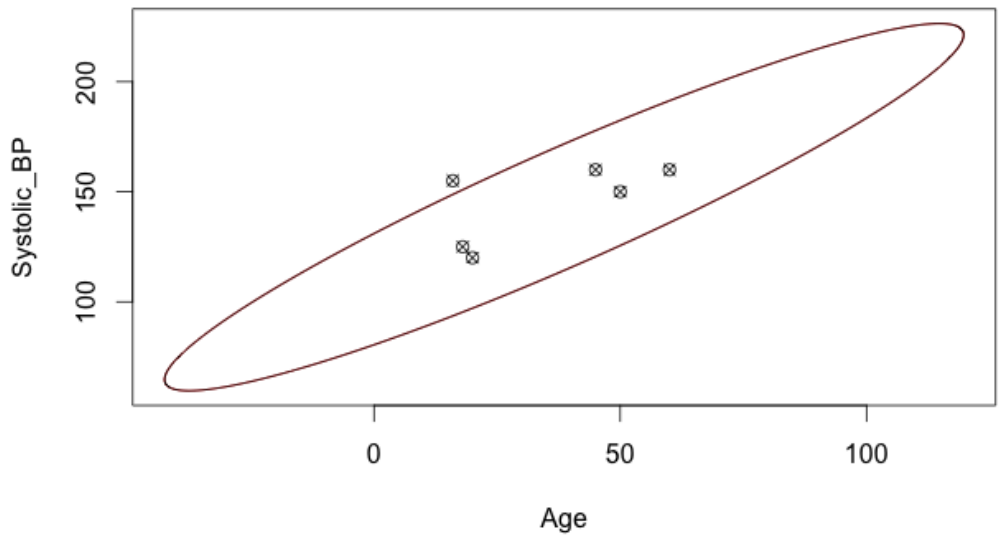


Figure 3.3: MCD Outlier

# Chapter 4

# Confidence Interval, Bootstrapping, and Effect Size

## 4.1 Confidence Interval

Perhaps you heard this statement that confidence intervals are always preferred over point estimates. Why is this statement true? A simple answer is that if you run any study you obtain one value of sample statistic (say mean $M_1$) out of potentially infinite values which you could obtain if you keep running your study infinite number of times. This brings up the question of how much faith can I have in $M_1$ that you obtained from your one study? Let's imagine I have access to the true population mean $\mu$. In this event, I can compare your $M_1$ with $\mu$ and make an educated guess about how correct you were in your estimate of $M_1$. However, there is a little problem, we invariably never know $\mu$. This makes the task of understanding how close your obtained $M_1$ is to $\mu$ very difficult. This is where the Confidence Interval (CI) comes to the rescue. One way to think of the 95% CI obtained from a study is as the interval in which the true population parameter actually lies. Let's take a specific example, if your $M_1$ = 4.5 and the 95% CI is $[3.25, 5.75]$ then the CI simply shows that with 95% certainty we can say that the true value $\mu$ lies in that interval [1].

Calculation of the confidence intervals are based on the central limit theorem. We will try to understand this by taking the example of calculating the confidence interval around sample means. The central limit theorem says that the sample mean is normally distributed in large samples (generally of size > 30). This implies that if we draw enough samples and obtain means of each of those samples, these means will be normally distributed. Formally, if $\bar{X}$ is the sample mean then

$$\bar{X} \sim N(\mu, \frac{\sigma}{\sqrt{n}}).$$

This means that sample means are distributed normally with mean $\mu$ (i.e., true population mean) and standard deviation $\frac{\sigma}{\sqrt{n}}$ (here $\sigma$ is the population standard deviation).

[1] There is a logical fallacy in saying that there is 95% probability that the true mean $\mu$ would fall in this interval. This statement would imply that the true mean is not a constant (in reality it is a constant). If it is a constant then its probability to fall in any interval can either be 0 (no it will not fall) or 1(yes it will fall). See Cumming and Finch (2005).

The fun part is that we don't know $\mu$ and $\sigma$, we only know that $\bar{X}$ is distributed normally. But we know a lot about how the normal distribution works. For instance, the sample mean can be rescaled to a standard normal:

$$\frac{\bar{X} - \mu}{\sigma/\sqrt{n}} \sim N(0,1)$$

It is quite easy to find out the the interval that contains the standard normal variable with 95% probability. We know from the properties of standard normal distribution that this interval is [-1.96, 1.96]. This means

$$Pr(-1.96 < \frac{\bar{X} - \mu}{\sigma/\sqrt{n}} < 1.96) = .95$$

Simple algebraic changes would yield

$$Pr(\bar{X} - 1.96\frac{\sigma}{\sqrt{n}} < \mu < \bar{X} + 1.96\frac{\sigma}{\sqrt{n}}) = .95$$

if we use $s$ (sample standard deviation) as a proxy for $\sigma$ then 95% confidence interval is

$$[\bar{X} - t_{.95}(n-1)\frac{s}{\sqrt{n}}, \bar{X} + t_{.95}(n-1)\frac{s}{\sqrt{n}})].$$

$\frac{s}{\sqrt{n}}$ is also known as standard error. $t_{.95}(n-1)$ is at $\alpha = .05$ level critical value of $t$ distribution with $n-1$ degrees of freedom. As this formula shows, the width of the CI is inversely proportional to $n$ (as sample size increases, width decreases) and directly proportional to the sample standard deviation ($s$).

### 4.1.1 Caution with the CI

1. Due to the Central Limit Theorem, the sample size has to be large (i.e., n > 30).
2. When the sample size is 8 to 29, use a normal probability plot to see if the data comes from a normal distribution. If it does not violate the normality assumption, use the confidence interval.

## 4.2 Effect Size

We briefly discussed the concept of effect size in our discussion of Power analysis and sample size determination. Effect size is essentially a way to quantify the difference between two groups. Revisiting our example of the price and quantity discount study, effect size would quantify how much willingness to buy increases with price discount compared to quantity discount. Since a study can have any other scale to measure willingness to buy (e.g.,1-7, 1-10, 0-100), effect size standardize the difference so results can be compared across different studies.

Historically, effect size appeared in meta analysis, however, it is crucial for any empirical investigation.

### 4.2.1 A Simple Example

One of the simplest ways to understand effect size is to consider Cohen's $d$. If mean and standard deviation of group 1 is $\bar{x}_1$, $s_1$ respectively and group 2 is $\bar{x}_2$, $s_2$. Sample size of group 1 and group 2 is $n_1$ and $n_2$ respectively then Cohen's $d$ is

$$d = \frac{\bar{x}_1 - \bar{x}_2}{s}$$

where $s$ is the pooled standard deviation

$$s = \sqrt{\frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}}$$

As these formulae show, the calculation of effect size is very straightforward where unlike power we always have a formula. Effect size is always unit independent.

### 4.2.2 Effect Size for factorial ANOVA

If you have a standard 2 x 2 between participants design then there are several options to calculate effect size. Here we will discuss one which can be computed by hand. It is partial $\eta^2$ ($\eta_p^2$). Let's assume we have Gender and Condition as our independent variables and the following is an AVOVA table.

| | Df | Sum Sq(SS) | Mean Sq | F value | Pr(>F) |
|---|---|---|---|---|---|
| Condition | 2 | 21715.02 | 10857.51 | 1.76 | 0.2078 |
| Gender | 1 | 10820.59 | 10820.59 | 1.76 | 0.2064 |
| Condition X Gender | 2 | 81979.56 | 40989.78 | 6.65 | 0.0093 |
| Residuals/Error | 14 | 86287.78 | 6163.41 | | |

$$\eta^2_{Pcondition} = \frac{SS_{condition}}{SS_{condition} + SS_{error}} = \frac{21715.02}{21715.02 + 86287.78} = 0.2$$

$$\eta^2_{Pgender} = \frac{SS_{gender}}{SS_{gender} + SS_{error}} = \frac{10820.59}{10820.59 + 86287.78} = 0.11$$

$$\eta^2_{Pcondition:gender} = \frac{SS_{condition:gender}}{SS_{condition:gender} + SS_{error}} = \frac{81979.56}{81979.56 + 86287.78} = 0.48$$

### 4.2.3 Effect Size for categorical data

Imagine you had 200 people who were randomly assigned to condition1 or condition2. In each condition they can choose from brand1, brand2 or brand3. Here are the results

| | Brand1 | Brand2 | Brand3 |
|---|---|---|---|
| Condition 1 | 30 | 10 | 60 |
| Condition 2 | 35 | 30 | 35 |

$\chi^2$ test provides the following results: $\chi^2(2) = 16.963, p < .0002$. The effect size in such situations is Cramer's V where

$$V = \sqrt{\frac{\chi^2/n}{\min(k-1, r-1)}}.$$

In our example, $n = 200$, $k = 3$ and $r = 2$ therefore

$$V = \sqrt{\frac{\chi^2/n}{\min(k-1, r-1)}} = \sqrt{\frac{16.963/200}{\min(3-1, 2-1)}} = \sqrt{\frac{0.084}{1}} = .291$$

Here is an *R* code that computes Cramer's V

```
require(lsr)
condition1 <- c(30, 10, 60)
condition2 <- c(35, 30, 35)
X <- cbind( condition1, condition2 )
# To test the null hypothesis, we would run a chi-square test:
chisq.test(X)
# Now we calculate Cramer's V:
cramersV( X )
```

## 4.3 Confidence Interval around Effect Size

From our discussion so far, Effect size just like the mean is a point estimate (i.e., it is just one value), it provides no idea about variability if a similar study was repeated. In other words, point estimate of effect size does not provide us any information about the range that may cover the true population effect size.

Just like the above discussion of confidence interval of means, now we have to understand how we can create 95% confidence interval around effect size. For means the process of applying confidence interval around mean estimate is quite simple since we use the common $t$ distribution. However, for effect size we use the non-central $t$ distribution [2]. For any two independent group $t$-test, you get $t$ value and degrees of freedom. If the non-centrality parameter for this t values and df is $ncp$ (if you want to learn how to calculate it by hand see [3] ), then the 95% confidence interval around Cohen's $d$ is

$$P\left(\frac{ncp_l}{\sqrt{\frac{n_1 n_2}{n_1+n_2}}} \leq d \leq \frac{ncp_u}{\sqrt{\frac{n_1 n_2}{n_1+n_2}}}\right) = .95.$$

[2] The noncentrality parameter is the normalized difference between $\mu_0$ and $\mu$. The noncentral $t$ distribution shows how the $t$ test statistic is distributed when the alternative hypothesis is true (e.g., when $\mu_0 \neq \mu$).

[3] David C. Howell's note on Confidence Intervals on Effect Size: `http://tinyurl.com/q6zsq6g`

Here $ncp_l$ and $ncp_u$ are lower and upper cutoff of the non-centrality parameter.

Most of the time you won't have to perform these calculations by hand, we will use an *R* code to understand how to do this. If we look at the results of our price and quantity discount study, we find that $n_1 = 30, n_2 = 30, Mean_P = 5.9, SD_p = 1, Mean_Q = 4.9, SD_Q = 1, t = 3.8730$. From above formulae Cohen's $d$ =1 (from Cohen's d formula). Now let's calculate 95% CI around this point estimate (first install MBESS and compute.es packages in R):

```
require(MBESS)
d = 1 # Cohen's d
n1 = 30 # Sample Size Price Condition
n2 = 30 # Sample Size quantity Condition
jj =conf.limits.nct(ncp=3.8730, df=58, conf.level=.95) # calculates lower and
    upper limits on ncp

jj$Lower.Limit/sqrt(n1*n2/(n1+n2)) # Lower bound on Cohen's d
jj$Upper.Limit/sqrt(n1*n2/(n1+n2)) # Upper bound on Cohen's d
```

or you can simply run the following code with means,(m.1=5.9, m.2=4.9), sample sds(sd.1=1, sd.2=1), and sample sizes(n.1= 30, n.2= 30):

```
require(compute.es)
mes(m.1=5.9, m.2=4.9, sd.1=1, sd.2=1, n.1=30, n.2=30, level = 95, cer = 0.2,
    dig = 2, verbose = TRUE, id=NULL, data=NULL) #Calculates Cohen's d and its
    95% confidence interval
```

If data is normally distributed, these CIs (around means, effect size etc.) are accurate. However, traditionally calculated CIs are not robust to deviations from normality. So in situations when data contains outliers that are skewing the sample mean and standard deviation, or when data is not normally distributed a better approach is to use bootstrap CIs for means, effect size etc. First, let's quickly understand how bootstrapping works.

## 4.4 A Short introduction to Bootstraping

This is a really short introduction to Bootstrapping. But Bootstrapping is an extensive topic. Those interested should read Efron and Gong (1983); Efron and Tibshirani (1994). To give you a basic idea of what bootstrapping is, we use one of the simplest bootstrap method- the percentile method.

Let's revisit the price/quantity discount example discussed earlier. Imagine we ran a study to find out if 20% price discount is more preferred than 20% bonus quantity. We chose soap as our stimulus and recruited 60 participants. 30 were randomly assigned to the price discount condition (P condition) and 30 to quantity discount condition (Q condition). Our dependent variable was people's willingness to buy soap on a 1-7 scale. We found out that in the P condition the mean willingness to buy was 5.9 and in Q it was 4.9, $p < .05$.

Whenever we analyze results of a study like this, our objective is not just to state what we found in the study but also to generalize results to its parent population. Sample statistics (mean, median, sample SD) obtained in any study are not stable, they

change if we draw another sample. For instance, if we run another identical study on price/quantity discount with a different sample (while keeping the sample size the same), it is highly unlikely that we will again obtain similar mean willingness to buy. Let's assume that we found mean willingness to buy in P condition ($M_P$ = 5.7). In the first trial we found $M_P$ = 5.9, in the second trial we found $M_P$ = 5.7. Now the question comes, what is the magnitude of such fluctuations. One simple answer to this question is running identical studies many many times and plotting sample means obtained in each such study. As you can imagine, it is impossible to run so many studies. To understand fluctuations, the second option is to delve into mathematical statistics and understand properties of sampling distributions. In contrast to these two approaches bootstrapping offers a very interesting solution.

We have 30 data points from the first trial when we obtained $M_P$ = 5.9. Let's assume this is our proxy population. Now resample with replacement from this proxy population. Here is an example that illustrates this process. If $X = \{5.2, 3.5, 6.8, 5.1, .....\}$ contains all the data points from the price condition in our first trial. In resampling with replacements, we will first randomly pick a point from $X$, write down its values and put it back in $X$. We then pick a second point, write down its value and put it back in $X$. We repeat this process for 30 times. Now we have our first bootstrap sample. We can calculate its mean $M_P^{B1}$, here $B1$ denotes the first bootstrap sample. Let's assume we repeat the whole process 10000 times. This will give us ten thousand values of mean $(M_P^{B1}, M_P^{B2}, M_P^{B3} .....M_P^{B10000})$. If we remove 2.5% of the highest values and 2.5% of the lowest values of mean, we will get 95% bootstrap confidence interval (CI) around our observed mean $M_P$ = 5.9.

Let's understand this process by a simple simulation in R.

```
require(boot)
n= 30 # sample size
m= 5.9 # sample mean
sd = 1 # sample SD
data <- round(rnorm(n, m, sd)) # generate sample
B=10000 # number of bootstrap samples
resamples <- lapply(1:B, function(i)
  sample(data, replace = T))
r.mean <- sapply(resamples, mean) # mean can be changed to median
se= sd(r.mean)
interv= mean(r.mean)+c(-1,1)*2*se
cat ("95% bootstrap confidence interval:", interv)

hist(r.mean, main="Bootstrap Mean Distribution",xlab="Bootstrap Means",
    ylab="")
```

This code will give 95% bootstrap confidence interval (for this particular sample it was 5.74, 6.46) and print the following histogram (see figure 4.1).

To get an intuitive understanding of how bootstrap CIs around the mean change, keep the mean ($m$) constant and change sample size ($n$) and sd in the above code. You will notice that as you increase sample size (i.e, collect more data in your studies), the CI becomes tighter. However, as you increase sd, the CI becomes wider.

Why bootstrap CIs are better? Bootstrap CIs are non parametric. Which means we don't need to make distributional assumptions about the data. Second, bootstrap CIs

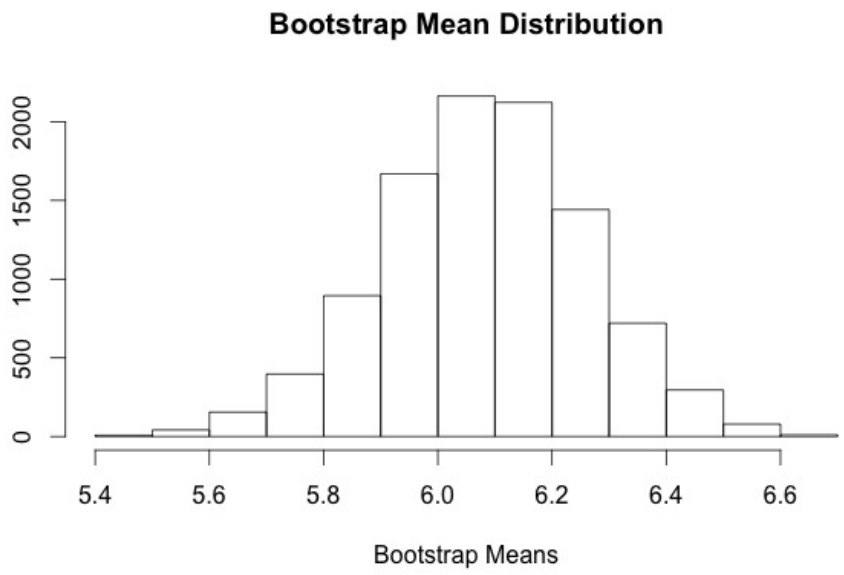


Figure 4.1: Bootstrap Means Distribution

are less susceptible to outliers. However, bootstrap CI is not a solution for small sample size. To apply bootstrap CI, you need to have at least 20 respondents per cell.

While we used the percentile bootstrap method to understand how bootstrapping works, this is not the most appropriate method to use in actual CI calculations. Bias-Corrected-and-Accelerated (BCa) method is more appropriate. Those interested in knowing how it works, read Efron (1987).

## 4.5 Bootstrap CI around Effect Size

We will use the bootES package of R to estimate CI's around effect size. It is strongly recommended that you read Kirby and Gerlanc (2013) before using these codes.

This examples uses a very simple data where females and males were asked to rate how much they like cheese pizza on a 1-10 scale. You can replace female, male with any manipulation you use in your studies.

```
require(bootES)
gender=c("f","m","f","m","f","m","f","m","f","m","f","m","f","m","f","m","f","m","f","m")
response=c(10,2,8,4,2,4,8,2,6,4,8,6,1,5,7,2,8,3,9,7)

data = data.frame(gender,response)

bootES( data, data.col = "response", group.col = "gender",
       contrast = c("m", "f"), effect.type = "cohens.d",ci.conf = 0.95,
           R=10000 ) #here R= 10000 means the code is generating 10000
           bootstrap samples
```

The output shows that the effect size (Cohen's $d$) for this data is 1.159 and 95% bootstrap CI is [.035, 2.862]. Since bootstrapping draws random samples, every time you run this

code the CI will be slightly differently.

# Chapter 5

# Quantile Regression

Here is an excerpt of the gender-based achievement gap in a math study described by Petscher and Logan (2014) that captures the essential idea of quantile regression:

It is fairly clear that math achievement should be predicted from gender rather than the reverse. The hypothesis tested by the authors' study was that the differences between males and females on math achievement may vary depending on how good the students are at math. Because females often choose not to take higher level math courses, the gender gap may be wider at higher levels of math achievement; however, the cut off point that will correspond to "high levels" of math achievement is unknown. The authors chose to use quantile regression because it allows for the estimation of the achievement gap between males and females at multiple points in the distribution of math achievement with no selected cut off points and no constraints on the functional form of the relation across the distribution of math achievement. Using this method, the authors identified that the achievement gap was near zero for low levels of math achievement, but was much larger at the higher end.

A similar situation arises when we try to understand the influence of BMI on running speed. It is quite intuitive that BMI is predictive of running speed, but it is also true that the relationship won't be as strong for the fastest runners. In other words, BMI's influence on running speed is different at different levels of running speed. Presumably BMI influences speed for those who are not in the top percentile in running speed differently from those in the lower percentile.

Both examples highlight that such questions cannot be answered by standard regression, because it predicts average relationship (i.e., 1 unit increase in BMI reduces running speed on an average by .25 miles/hour) as the relation is assumed to be the same for everyone.

What's wrong in segmenting our dependent variables and running separate linear regression on each segment. So for the BMI example it will mean segmenting running speed into various groups (according to its unconditional distribution) and then running separate linear regressions on them. Why can't we do that? The simple answer: any truncation of this sort will lead to sample selection problems where it will be hard to consider resulting subgroups of data as randomly selected. If you are interested in exploring this point further, please read Heckman's (1979) work on sample selection.

## 5.1 What is Quantile Regression?

**Warning:** Unlike topics discussed so far, Quantile regression requires the understanding of some slightly advanced statistical concepts. Subsequent sections provide some rudimentary description of such concepts. Ideally, you should read the first 5 chapters of Hao and Naiman (2007) to gain a holistic understanding of Quantile regression.

Classical regression (least square regression or LSR) focuses on the expectation of a variable $Y$ conditioned on a set of variables $X$, i.e., $E(Y|X)$. One of the main problems with this approach is that it restricts itself to just the conditional mean.

Quantile regression on the other hand let's us understand the conditional distribution of $Y$ (on $X$) at different locations (not just the mean).

Let's take a simple example: if we want to study the influence of family income on SAT scores, LSR

$$Y_{SAT} = \beta_0 + \beta_1 X_{income} + \epsilon$$

will help us get values of $\beta_0$ and $\beta_1$. $\beta_1$ will tell us how average SAT scores will change for every unit change in income. However, $\beta_1$ will not let us know how every unit change in income influences SAT scores of students who are in the top 10 percentile vs. bottom 10 percentile of SAT scores. In other words, LSR will tell us how on average SAT scores change with change in income but will not help us understand the role income plays in groups of worst and best SAT performers. Quantile regression can help us answer that question. In addition, Quantile regression does not assume errors to be homoskedastic and is less susceptible to outliers. Let's understand what are quantiles.

### 5.1.1 Quantiles

A good starting point is the difference between mean and median.

Mean $\mu$ can be defined as

$\mu = \arg\min_c E(Y - c)^2$,

which basically states that for a random variable $Y$, mean is the center $c$ of a distribution which minimizes the squared sum of deviations.

Median $m$ also has a similar minimization property. Instead of minimizing squared distance we can use absolute distance $|Y - c|$ and minimize mean of $|Y - c|$ i.e.,

$m = \arg\min_c E|Y - c|$.

Applying the same line of reasoning for samples we can obtain $\hat{\mu}$ and $\hat{m}$. Median is a special case of quantiles (50% or .5 quantile). If we want to generalize the idea of median to other quantiles, we need to understand CDF (Cumulative Distribution Function).

$F_Y$ (CDF) of a random variable Y provides us for a given value of $y$, the proportion of the population for which $Y \leq y$. Formally

$F_Y(y) = F(y) = Pr(Y \leq y)$

Let's take an example. For standard normal distribution, $F_Y(0) = .5$, which means 50% of values lie below 0. Similarly $F_Y(1.28) = .9$. More generally we say $\theta$ quantile is value $y$ such that $Pr(Y \leq y) = \theta, \forall\theta \in [0,1]$. Therefore, $F_Y(y) = F(y) = \theta$.

Quantile function ($Q_Y(\theta)$) is the inverse of $F_Y$.

$Q_Y(\theta) = Q(\theta) = F_Y^{-1}(\theta) = \inf[y : F(y) \geq \theta]$

In words, this means $\theta^{th}$ quantile of a CDF $F$ is the minimum (infimum) of the set of values $y$ such that $F(y) \geq \theta$. As an example $F_Y(0) = .5$ means 0 is $50^{th}$ quantile because the set that contains values with $Pr[Y > 0]$ is large and $50^{th}$ quantile is minimum of all those values.

### 5.1.2 Estimating Quantile Regression

The easiest way to understand how quantile regression is estimated is to start with simpler least square regression. LSR can be defined as a minimization problem. The least-squares estimator finds for the parameter estimates $\hat{\beta}_0$ and $\hat{\beta}_1$ by considering those values of the parameters that minimize the sum of squared residuals, i.e,

$$\underset{\beta_0,\beta_1}{\arg\min} \sum_i (y_i - (\beta_0 - \beta_1 x_i)^2).$$

Extending this to median regression (a special case of quantile regression since median is 50% or .5 quantile) is straightforward. The aim is to find the coefficients that minimize the sum of absolute residuals (the absolute distance from an observed value to its fitted value). The estimator solves for $\hat{\beta}_0^{.5}$ and $\hat{\beta}_1^{.5}$ by minimizing

$$\underset{\beta_0^{.5},\beta_1^{.5}}{\arg\min} \sum_i |y_i - \beta_0^{.5} + \beta_1^{.5} x_i|.$$

More generally for the $p^{th}$ quantile the minimization problem is

$$\underset{\beta_0^p,\beta_1^p}{\arg\min}\ p \sum_{y_i \geq \beta_0^p + \beta_1^p x_i} |y_i - \beta_0^p + \beta_1^p x_i| + (1-p) \sum_{y_i < \beta_0^p + \beta_1^p x_i} |y_i - \beta_0^p + \beta_1^p x_i|$$

The proportion of data points lying below the fitted line $y = \hat{\beta}_0^p + \hat{\beta}_1^p x$ is $p$, and the proportion lying above is $1 - p$. In the simplest case if $p = .5$ then this fitted line represents the median regression line.

The minimization problem also shows that the estimation of coefficients for each quantile regression utilizes the weighted data of the whole sample, not just the portion of the sample at that quantile.

## 5.2 How to use and interpret quantile regression?

We will use the R package quantreg [1]. Let's use the Engle data supplied with the quantreg package. Engel data has 2 variables food expenditure and income per household. The idea is to find out how income changes money spent on food consumption. LSR will inform as how on an average food expenditure will change for every unit change in income. With quantile regression, we will get $\beta^p$ for each quantile which would tell us how at each quantile of food expenditure, income influences food consumption.

```
library(quantreg)
data(engel)
fit1 <- rq(foodexp ~ income, tau = c(.1,.3,.5,.7,.9), data = engel) # tau is
    quantile value, here it is from .1 to .9
summary(fit1, se = "nid") # to compute bootstrapped standard errors
plot(summary(fit1), nrow = 1, ncol = 2)
```

Let's look at the output for .5 and .9 quantiles. For the .5 quantile the results show:

0.5 Quantile Coefficients

| | Value | Std. Error | t value | Pr(> \|t\|) |
|---|---|---|---|---|
| Intercept | 81.48 | 25.22 | 3.23 | .0001 |
| income | .56 | .032 | 17.03 | .00001 |

For .5 quantile the coefficient estimate is interpreted as the change in the median of the response variable (foodexp) corresponding to a unit change in the predictor (income).

0.9 Quantile Coefficients

| | Value | Std. Error | t value | Pr(> \|t\|) |
|---|---|---|---|---|
| Intercept | 67.35 | 24.00 | 2.80 | .005 |
| income | .68 | .028 | 23.82 | .00001 |

Similar to median (i.e., .5 quantile), the .9 quantile coefficients show change in the .9 quantile of food expenditure corresponding to a unit change in the income.

Across these 2 tables the main takeaway is that unlike LSR, the coefficients are not the same for each quantile. For .9 quantile of food expenditure, the change in income has larger influence(.68) than at .5 quantile of food expenditure (.56).

Like $R^2$ for LSR, it is possible to calculate the Pseudo $R^2$ based on Koenker and Machado (1999) at the $p^{th}$ quantile. The formula is

$$R_1(p) = 1 - \frac{\sum_{y_i \geq \hat{y}_i} p \cdot |y_i - \hat{y}_i| + \sum_{y_i < \hat{y}_i} (1-p) \cdot |y_i - \hat{y}_i|}{\sum_{y_i \geq \bar{y}} p \cdot |y_i - \bar{y}| + \sum_{y_i < \bar{y}_i} (1-p) \cdot |y_i - \bar{y}|}$$

Here is a code to calculate Pseudo $R^2$ for Engel data

[1] See http://cran.r-project.org/web/packages/quantreg/quantreg.pdf and http://cran.r-project.org/web/packages/quantreg/vignettes/rq.pdf

```
require(quantreg)
data(engel)
fit0 <- rq(foodexp~1,tau=0.5,data=engel) # base model
fit1 <- rq(foodexp~income,tau=0.5,data=engel)
anova(fit1,fit0)
rho <- function(u,tau=.5)u*(tau - (u < 0))
R1 <- 1 - fit1$rho/fit0$rho
print(R1) # this is the value of Pseudo R squ at .5 quantile
```

# Appendix A

# Some Basic Concepts

## A.1 Fundamental Misconceptions about p-values

**Source: These misconceptions are compiled from sources around the web**

Q: If I am not able to reject the null hypothesis then it means I can accept it (example: ruling out non-favorite/alternate accounts with null results).

A: No. You are not able to reject null hypothesis because you have insufficient evidence. Many times with higher power you can reject any null hypothesis. So don't use inability to reject null (i.e., higher $p$ values) as a way to rule out your non-favorite/alternate accounts.

---

Q: $p$ values tell me the probability that the null hypothesis is incorrect.

A: No. $p$ values simply says that your data is more extreme if we assume $H_0$ is true.

---

Q: $p < .05$ is an objective standard to determine statistical significance.

A: This is one of the most unfortunate misconceptions. $p < .05$ is just a convention.

---

Q: if I get $p < .01$ then my evidence is better than if someone got $p < .05$.

A: No. $p$ values are meaningless to derive this conclusion. What you need is Effect Size.

---

Q: if I get $p = .05$ then it means that my results can be replicated 95% of the time.

A: No. $p$ value says absolutely nothing about replication. See the next topic.

---

## A.2 $p$-value and replication

Does $p$-value tells us anything about replicability? In its simplest definition it informs us that if we as researchers conducted an experiment testing for the null hypothesis $H_0$ versus $H_1$ (i.e. the hypothesis of interest to us) with a sample size of 100 and found

$p < .05$, then this would only inform us that the chances of finding the effect in the sample when $H_0$ was actually true in the population is only 5%. However, if $H_0$ were false, we cannot infer the replicability of the effect proposed by $H_1$ from the $p$-value. While conducting an experiment since a priori we do not know whether $H_0$ is true or not, it is difficult to infer replicability.

The problem lies in the erroneous beliefs that have cropped up in the literature for what $p$-value means. For instance, there exists the completely erroneous belief that when a researcher gets a $p < .04$ (against the null), it is equivalent to the probability of replicating the proposed effect with a 96% probability. Cautioning against such erroneous beliefs, Cumming (2008) states very clearly that $p$-value is a very unreliable measure of replication since it varies a lot across replications: ".....if an initial experiment results in two-tailed $p = .05$, there is an 80% chance the one-tailed $p$ value from a replication will fall in the interval (.00008, .44), a 10% chance that $p < .00008$, and fully a 10% chance that $p > .44$."

Killeen (2005) takes a Bayesian approach to argue that $p$-values are tests of the null-hypothesis $H_0$, not the alternate hypothesis. He argues that $p(H_0|x \geqslant D)$ (test of the null hypothesis, where $D$ is the data) does not indicate (as is commonly, erroneously, assumed) the reverse $p(x \geqslant D|H_0)$ (which would be the test of $H_1$). A Bayesian approach would yield the right answer if we could calculate the former using $p(H_0 : x \geqslant D) = p(x \geqslant D|H_0) * p(H_0)/p(x \geqslant D)$ . However, we would need to know the prior probabilities of both $H_0$ and $D$. These priors are generally unknowable leading us back to the same quandary of not being able to conclusively reject $H_0$ without knowing the priors and of also not being to accept the alternate. Finally, if we knew the effect size of the proposed effect (captured by $H_1$) in the population then we can predict the replicability of the experiment. However, we can only infer the effect size of the population through the effect size that we observed in the experiment. This again brings us back to dilemma of not knowing the priors.

## A.3 Descriptive vs Inferential Statistics

Yearly batting average of Babe Ruth is a descriptive statistic. We see such numbers in everyday life. Average rainfall in a specific city, crime rate etc.. Basically descriptive statistics is summary of data. It is quite straightforward and easy to understand.

If we want to test theories about the nature of the world in general based on samples taken from the world then we are dealing with inferential statistics. In other words inferential statistics helps us infer the population's characteristics from the sample's characteristics. Let's take a very simple example. If in a sample you find that people like red colored ice creams more than blue colored ice creams, clearly, you want to get some idea about how this pattern of preference will generalize to population. Inferential statistics will help you make this generalization.

## A.4 PDF vs CDF

A PDF answers the question: "How common are samples at exactly this value?" A CDF answers the question "How common are samples that are less than or equal to this value?" The CDF is the integral of the PDF.

for a continuous random variable X, we can define the probability that X is in $[a, b]$ as $P(a <= X <= b) = \int_a^b f(x)dx$. (integral) Where $f(x)$ is probability density function, which satisfies two properties $f(x) >= 0$ and $\int_{-\infty}^{+\infty} f(x)dx = 1$. a, b are real numbers. Probability distribution function defines the probability that $X <= a$ as $P(X <= a) = \int_{-\infty}^{a} f(x)dx$

# Bibliography


Cumming, G. (2008). Replication and p intervals: p values predict the future only vaguely, but confidence intervals do much better. *Perspectives on Psychological Science*, 3(4):286–300.

Cumming, G. and Finch, S. (2005). Inference by eye: confidence intervals and how to read pictures of data. *American Psychologist*, 60(2):170.

Efron, B. (1987). Better bootstrap confidence intervals. *Journal of the American statistical Association*, 82(397):171–185.

Efron, B. and Gong, G. (1983). A leisurely look at the bootstrap, the jackknife, and cross-validation. *The American Statistician*, 37(1):36–48.

Efron, B. and Tibshirani, R. J. (1994). *An introduction to the bootstrap*. CRC press.

Hao, L. and Naiman, D. Q. (2007). *Quantile regression*. Number 149. Sage.

Hoenig, J. M. and Heisey, D. M. (2001). The abuse of power. *The American Statistician*, 55(1).

Killeen, P. R. (2005). An alternative to null-hypothesis significance tests. *Psychological science*, 16(5):345–353.

Kirby, K. N. and Gerlanc, D. (2013). Bootes: An r package for bootstrap confidence intervals on effect sizes. *Behavior research methods*, 45(4):905–927.

Koenker, R. and Machado, J. A. (1999). Goodness of fit and related inference processes for quantile regression. *Journal of the american statistical association*, 94(448):1296–1310.

Liu, X. S. (2013). *Statistical power analysis for the social and behavioral sciences: basic and advanced techniques*. Routledge.

Maxwell, S. E., Kelley, K., and Rausch, J. R. (2008). Sample size planning for statistical power and accuracy in parameter estimation. *Annu. Rev. Psychol.*, 59:537–563.

Rousseeuw, P. J. (1984). Least median of squares regression. *Journal of the American statistical association*, 79(388):871–880.

Rousseeuw, P. J. and Driessen, K. V. (1999). A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 41(3):212–223.