Week 2 Lab

Yuanchao Zhang

2018-02-11

Plan for today

1. Changes of lab session administration
2. Recap
3. Install packages
4. The normal distribution
5. Check normality
6. One-sample tests of proportion

Changes of lab session administration

Changes of lab session administration

- Lab handout:
    - Merged to homework assignment.
    - Written mostly by instructors.
- Optional sections of lab handout:
    - Separated from the handout & homework assignment.
    - Will show up in the ”Pages” section of Canvas.

Recap

Recap

- R environment
  - Associated with each R interpreter session
- R current working directory
  - Do NOT change during analysis
- Path
  - Absolute path
  - Relative path
  - Path delimiter:
    - `\` in Windows
    - `/` in Linux and MacOS
    - `/` in R works for Windows, Linux, and MacOS

Recap

- Tutorial of R data analysis
  - Input:
    - `read.csv()`
  - Analyze:
    - Plotting: `ggplot2`
    - Data manipulation: indexing, `subset()`, and `dplyr`
    - Statistical testing: `t.test()`
  - Output: `write.csv()` and `pdf()`

Install packages

Install packages

```r
# For normality test
install.packages('nortest')

# For power analysis of one sample proportion test
install.packages('pwr')

# You do not need to run this if you already
# installed `ggplot2'
install.packages('ggplot2')
```

The normal distribution

The normal distribution

When a random variable X is dristributed normally with mean μ and variance σ^2, we write:

$$X \sim \mathcal{N}(\mu,\, \sigma^2)$$

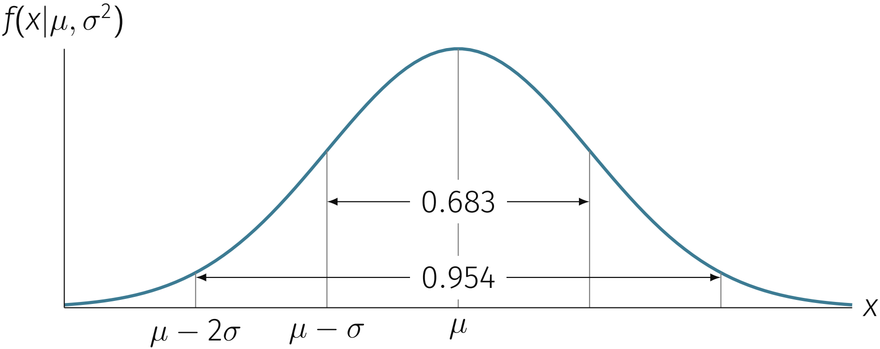
Probability density distribution (PDF) of the normal distribution:

$$f(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Source: Normal Distribution at en.wikipedia.org

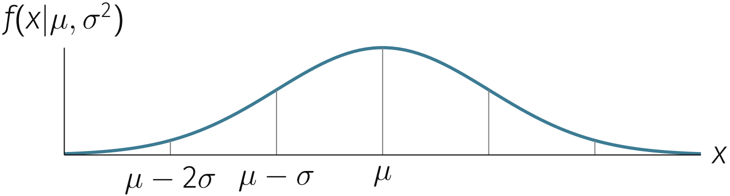
PDF of the normal distribution

$$f(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



Source: Jake at TeXExchange

The normal distribution

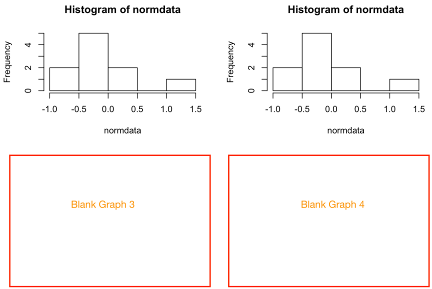


- Single Peaked
- Symmetric
- $E[x] = \mu$
- $Var[x] = \sigma^2$
- $SD[x] = \sigma$

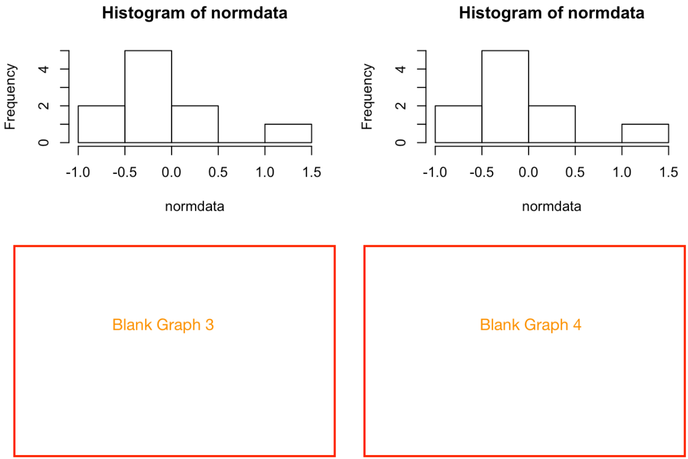
Source: Paul E. Johnson. CLT. 2011

Generate normal random numbers with `rnorm`

```
opar <- par(no.readonly = TRUE)
par(mfrow=c(2, 2))
normdata <- rnorm(10, mean=0, sd=1)
hist(normdata)
hist(normdata)
par(opar)
```

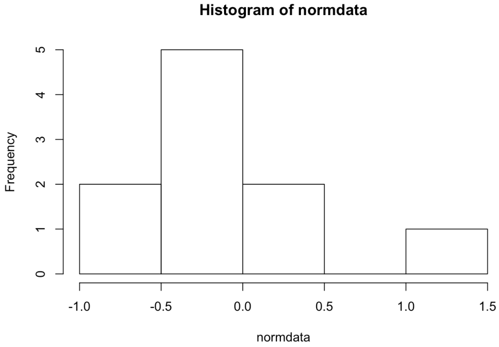


```
par(mfrow=c(2, 2))
```



par(opar)

```r
# Plotting parameters have been set to old ones
# in `opar'
hist(normdata)
```

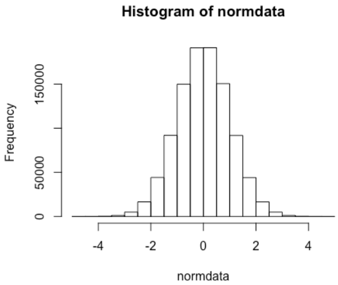


Generate normal random numbers with `rnorm`

```
normdata <- rnorm(10, mean=0, sd=1)
mean(normdata)
## [1] -0.1397156
sd(normdata)
## [1] 0.5693181
```

Generate 1 million normal random numbers

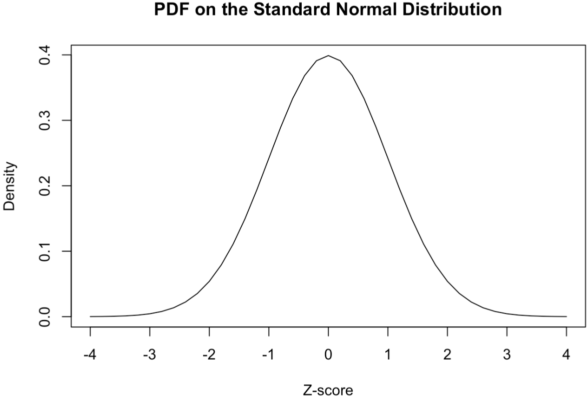
```
normdata <- rnorm(1000000, mean=0, sd=1)
mean(normdata)
## [1] -9.904492e-05
sd(normdata)
## [1] 1.000252
hist(normdata)
```



Calculate probability density of normal distribution with `dnorm`

```r
z_score_range <- seq(-4, 4, by = 0.2)
den_zscores <- dnorm(z_score_range)
plot(den_zscores, type = "l",
     main = "PDF on the Standard Normal Distribution",
     xlab = "Z-score", ylab = "Density", xaxt = "n")

den_zscore_sigmas <- c(dnorm(4), dnorm(3), dnorm(2),
                       dnorm(1), dnorm(0), dnorm(1),
                       dnorm(2), dnorm(3), dnorm(4))

den_score_labels <- c(-4, -3, -2, -1, 0, 1, 2, 3, 4)
axis(1, at = which(den_zscores %in% den_zscore_sigmas),
     labels = den_score_labels)
```

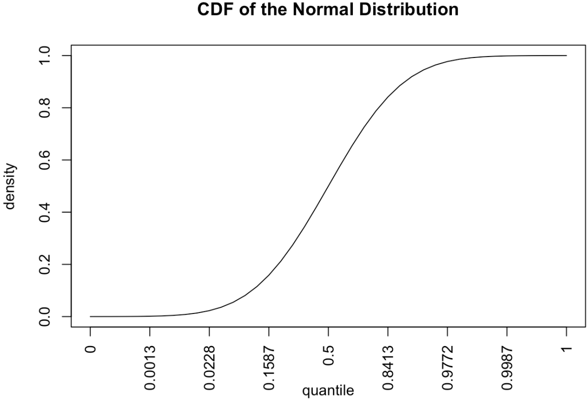
Calculate probability density of normal distribution with `dnorm`



Calculate cumulative probability of normal distribution with `pnorm`

```r
pnorm_scores <- pnorm(z_score_range)
plot(pnorm_scores, type = "l",
     main = "CDF of the Normal Distribution",
     xlab = "quantile", ylab = "density", xaxt = "n")

pnorm_values <- c(pnorm(-4), pnorm(-3), pnorm(-2),
                  pnorm(-1), pnorm(0), pnorm(1),
                  pnorm(2), pnorm(3), pnorm(4))

axis(1, at = which(pnorm_scores %in% pnorm_values),
     labels = round(pnorm_values, 4), las = 3)
```

Calculate cumulative probability of normal distribution with `pnorm`



Calculate quantiles of normal distribution with `qnorm`

```r
qnorm(0.99)
## [1] 2.326348

qnorm(0.9999)
## [1] 3.719016

pnorm(qnorm(0.9999))
## [1] 0.9999

# lower.tale: if TRUE (default), probabilities are P[X ≤ x]
# otherwise, P[X > x].
qnorm(0.9999, lower.tail = FALSE)
## [1] -3.719016

qnorm(1e-04)
## [1] -3.719016

pnorm(qnorm(0.9999, lower.tail = FALSE))
## [1] 1e-04
```

Central limit theorem

Theorem

Let $X_1, X_2, ..., X_n$ be a sequence of identically distributed (i.i.d.) random variables with mean $E[X_i] = \mu$ and finite variance $Var(X_i) = \sigma^2$. Define $S_n = \dfrac{1}{n}\sum_i X_i$. Then, as $n \to \infty$, $S_n \xrightarrow{\mathcal{D}} \mathcal{N}\left(\mu, \sigma^2/n\right)$.

Central limit theorem is difficult to prove algebraically, but it is quite easy to demonstrate with simulation.

Source: Central Limit Theorem at en.wikipedia.org

Demonstrate central limit theorem with simulation

Theorem

Let $X_1, X_2, ..., X_n$ be a sequence of i.i.d. random variables with mean $E[X_i] = \mu$ and finite variance $Var(X_i) = \sigma^2$. Define $S_n = \frac{1}{n}\sum_i X_i$. Then, as $n \to \infty$, $S_n \xrightarrow{\mathcal{D}} \mathcal{N}\left(\mu, \sigma^2/n\right)$.

- Simulation procedure:
  1. Let $X_1, X_2, ..., X_n$ be a sequence of i.i.d. uniformly distributed random variables.
  2. Calculate sample everage S_n.
  3. Repeat m times.
  4. Plot the empirical PDF of S_n and normal PDF stated by the theorem.

Demonstrate central limit theorem with simulation

```
# Let X1, X2, ..., X10 be a sequence of i.i.d.
# random variables uniformly distributed
# from 0 to 1.
runif(n = 10, min = 0, max = 1)
##  [1]  0.89804708 0.85956656 0.63841285
##  [4]  0.59958342 0.68575572 0.28295208
##  [7]  0.42904749 0.15577247 0.26559273
##  [10] 0.01854827
```

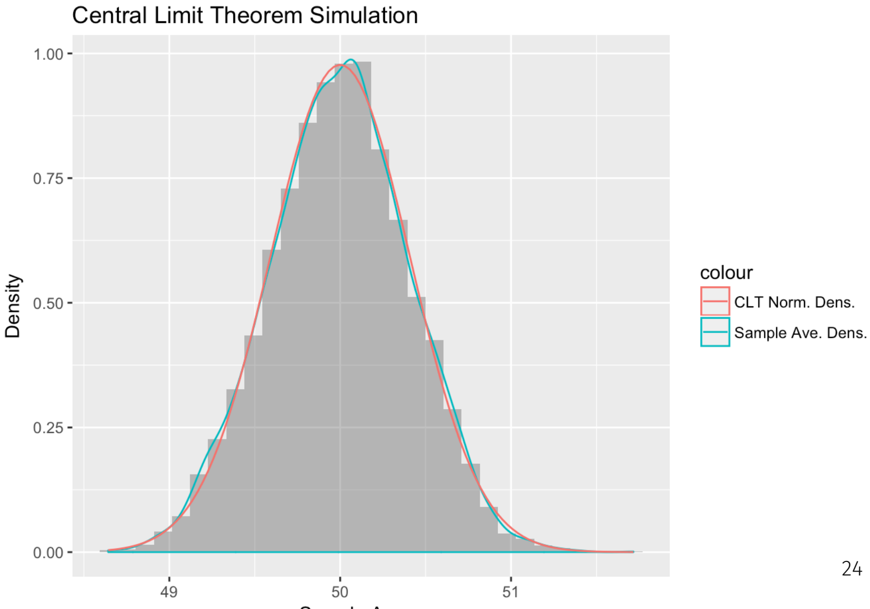
Demonstrate central limit theorem with simulation

```r
number_of_samples <- 5000
size_of_each_sample <- 5000
unif_min <- 0
unif_max <- 100

unif_mean <- (unif_max - unif_min) / 2
unif_sd <- (((unif_max - unif_min) ^ 2) / 12) ^ 0.5

sample_average_vector <- replicate(number_of_samples, {
  mean(runif(n = size_of_each_sample,
             min = unif_min,
             max = unif_max))
})
```

Demonstrate central limit theorem with simulation

```
clt_norm_mean <- unif_mean
clt_norm_sd <- unif_sd / (size_of_each_sample ^ 0.5)

ggplot(data = data.frame(x = sample_average_vector),
       mapping = aes(x = x)) +
  geom_histogram(mapping = aes(y = ..density..),
                 alpha = 0.4) +
  geom_density(mapping = aes(color = 'Sample Ave. Dens.')) +
  stat_function(fun = dnorm,
                args = list(mean=clt_norm_mean,
                            sd=clt_norm_sd),
                aes(colour = 'CLT Norm. Dens.')) +
  labs(x = 'Sample Average', y = 'Density') +
  ggtitle("Central Limit Theorem Simulation")
```

Demonstrate central limit theorem with simulation



Check normality

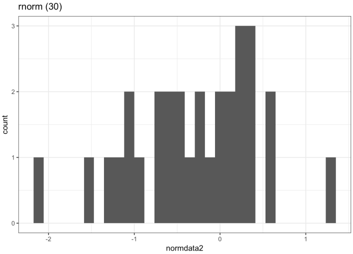
Methods to check normality

- Explorative data analysis (EDA)
    - Histogram
    - Density plot
    - Quantile-Quantile plot
- Statistical tests
    - Shapiro-Wilk normality test
    - Anderson-Darling normality test

Use EDA to check normality

Histogram:

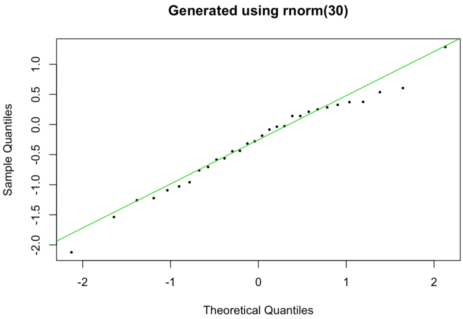
```
num_points <- 30
normdata2 <- rnorm(num_points)

qplot(normdata2) +
  geom_histogram() +
  theme_bw() +
  ggtitle(paste("rnorm (", num_points, ")", sep=""))
```



Use EDA to check normality

Quantile-Quantile (Q-Q) plot:

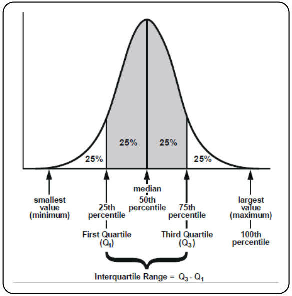
```
qqnorm(normdata2, pch=16, cex=0.5,
       main=paste("Generated using rnorm(",
                  num_points, ")", sep=""))
qqline(normdata2, col=3)
```

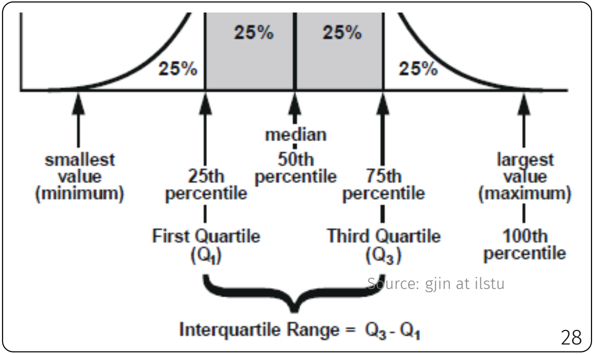


Q-Q plot

Quantiles divide probility distribution or sample observations into even intervals.

Example, 4-quantiles (quartiles) and 100-quantiles (percentiles):



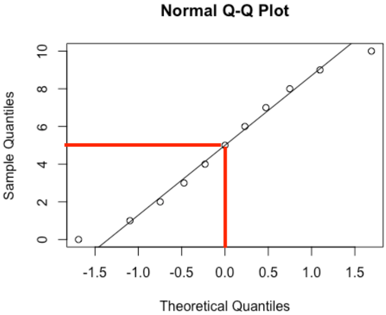


Source: gjin at ilstu

Q-Q plot

Q-Q plot compares the quantiles of one sample or distribution against the other.

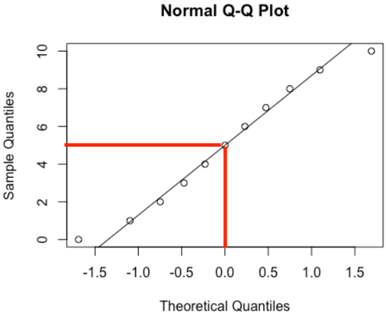
```
quantile(0:10, probs=seq(0, 1, 0.1))
##   0%  10%  20%  30%  40%  50%  60%  70%  80%  90% 100%
##    0    1    2    3    4    5    6    7    8    9   10
qqnorm(0:10, ylim = c(0,10))
qqline(0:10, ylim = c(0,10))
```



Q-Q plot

Q-Q plot compares the quantiles of one sample or distribution against the other.

```
quantile(0:10, probs=seq(0, 1, 0.1))
##    0%   10%   20%   30%   40%   50%   60%   70%   80%   90%  100%
##     0     1     2     3     4     5     6     7     8     9    10
qqnorm(0:10, ylim = c(0,10))
qqline(0:10, ylim = c(0,10))
```

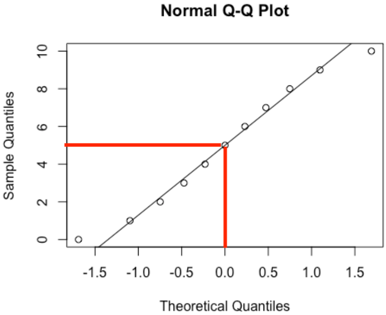


Anything wrong in this Q-Q plot?

Q-Q plot

Q-Q plot compares the quantiles of one sample or distribution against the other.

```
quantile(0:10, probs=seq(0, 1, 0.1))
##    0%   10%   20%   30%   40%   50%   60%   70%   80%   90%  100%
##     0     1     2     3     4     5     6     7     8     9    10
qqnorm(0:10, ylim = c(0,10))
qqline(0:10, ylim = c(0,10))
```



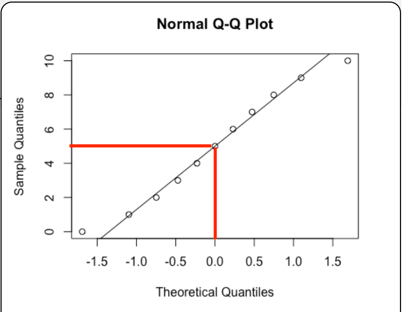
Anything wrong in this Q-Q plot?

Why the 0% of normal distribution is plotted around −1.75?

Normal quantile estimation

Source code of `qqnorm`:

```
getAnywhere(qqnorm.default)
## ...
##     x <- qnorm(ppoints(n))[order(order(y))]
## ...
seq0to10_qqnormplot <- qqnorm(0:10)
seq0to10_qqnormplot$x
## [1] -1.69 -1.10 -0.75 -0.47 -0.23  0.00
## [7]  0.23  0.47  0.75  1.10  1.69
seq0to10_qqnormplot$y
## [1]  0  1  2  3  4  5
## [6]  6  7  8  9 10
```



Normal quantile estimation

Source code of `ppoints`:

```
getAnywhere(ppoints)
## ...
##          (1L:n - a)/(n + 1 - 2 * a)
## ...
```

Source: Aniko at StatsExchange

Normal quantile estimation

Source code of `ppoints`:

```
getAnywhere(ppoints)
## ...
##         (1L:n - a)/(n + 1 - 2 * a)
## ...
```

Blom (1958): standard normal random variable

$E(r:n) \approx \Phi^{-1}(\frac{r-\alpha}{n-2\alpha+1})$ with $\alpha = 0.375$.

Source: Aniko at StatsExchange

Normal quantile estimation

Source code of `ppoints`:

```
getAnywhere(ppoints)
## ...
##          (1L:n - a)/(n + 1 - 2 * a)
## ...
```

Blom (1958): standard normal random variable $E(r:n) \approx \Phi^{-1}(\frac{r-\alpha}{n-2\alpha+1})$ with $\alpha = 0.375$.

In plain words: if you draw *n* standard normal random numbers and order them from lowest to highest, the *r*th number is mostly likely to be the value where the CDF has value $\frac{r-\alpha}{n-2\alpha+1}$.

Source: Aniko at StatsExchange

Normal quantile estimation

Source code of `ppoints`:

```
getAnywhere(ppoints)
## ...
##          (1L:n - a)/(n + 1 - 2 * a)
## ...
```

Blom (1958): standard normal random variable $E(r:n) \approx \Phi^{-1}(\frac{r-\alpha}{n-2\alpha+1})$ with $\alpha = 0.375$.

In plain words: if you draw *n* standard normal random numbers and order them from lowest to highest, the *r*th number is mostly likely to be the value where the CDF has value $\frac{r-\alpha}{n-2\alpha+1}$.

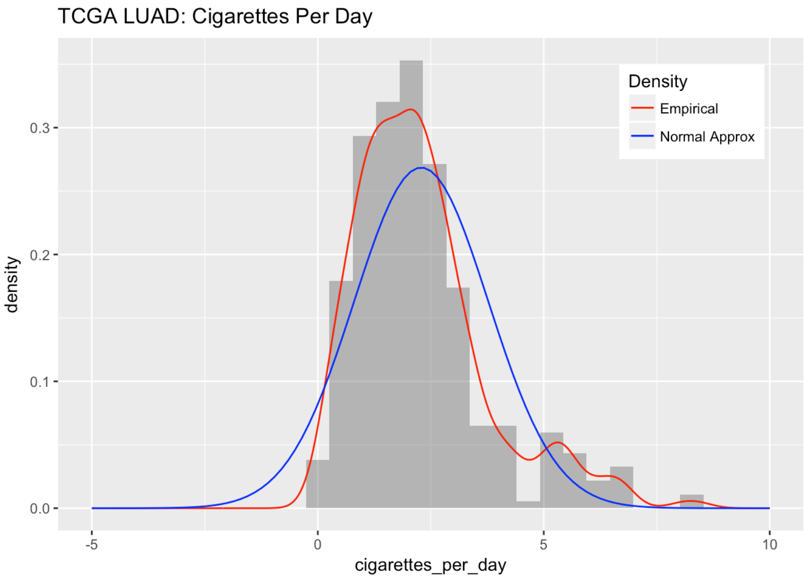
For more details, take STAT512 Mathematical Statistics.

Source: Aniko at StatsExchange

Histogram of TCGA-LUAD cigarettes per day

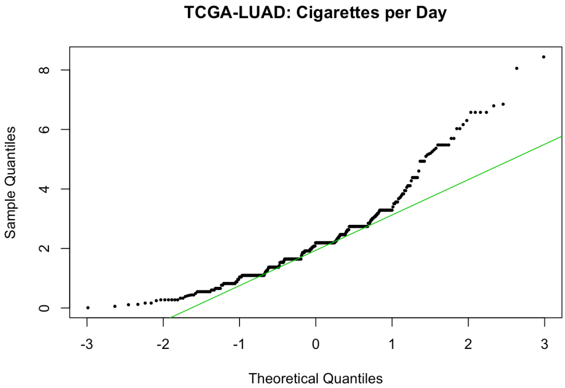
```r
luad_cpd_sd <- sd(tcga_luad$cigarettes_per_day,
                  na.rm=TRUE)
luad_cpd_mean <- mean(tcga_luad$cigarettes_per_day,
                      na.rm=TRUE)

qplot(x = cigarettes_per_day, xlim=c(-5, 10),
      data = tcga_luad, geom = "blank") +
  geom_histogram(aes(y = ..density..),
                 alpha = 0.4) +
  geom_line(aes(y = ..density.., colour = 'Empirical'),
            stat = 'density') +
  stat_function(fun = dnorm,
                args = list(mean=luad_cpd_mean,
                            sd=luad_cpd_sd),
                aes(colour = 'Normal Approx'))  +
  scale_colour_manual(name = 'Density',
                      values = c('red', 'blue')) +
  theme(legend.position = c(0.85, 0.85)) +
  ggtitle("TCGA LUAD: Cigarettes Per Day")
```

Histogram of TCGA-LUAD cigarettes per day

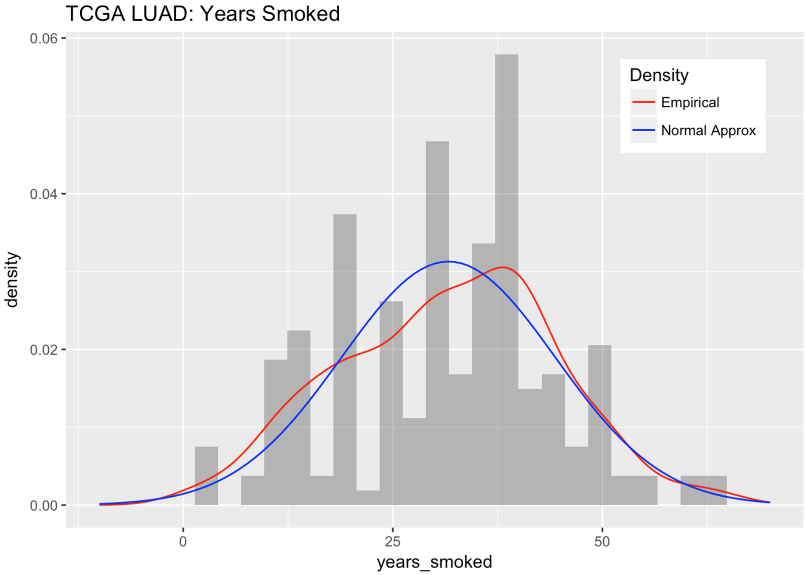


Q-Q plot of TCGA-LUAD cigarettes per day

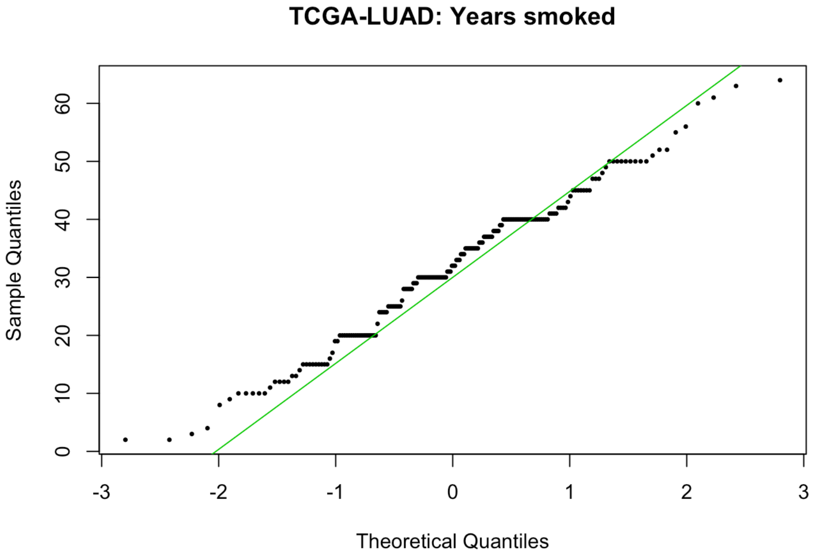
```
qqnorm(tcga_luad$cigarettes_per_day,
       pch=16, cex=0.5, main="TCGA-LUAD: Cigarettes per Day")
qqline(tcga_luad$cigarettes_per_day, col=3)
```



Histogram of TCGA-LUAD years smoked



Q-Q plot of TCGA-LUAD years smoked



Use statistical tests to check normality

Shapiro-Wilk normality test:

```
nortest_points <- 100
normdata_nortest <- rnorm(nortest_points)
shapiro.test(normdata_nortest)
##
##  Shapiro-Wilk normality test
##
## data:  normdata_nortest
## W = 0.97583, p-value = 0.06271
shapiro.test(1:1000)
##
##  Shapiro-Wilk normality test
##
## data:  1:1000
## W = 0.95481, p-value < 2.2e-16

# Smaller p-value indicates significant deviation from normal
# distribution.
```

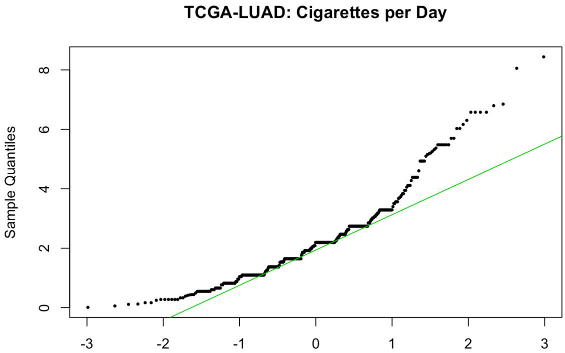
Use statistical tests to check normality

Anderson-Darling normality test

```
ad.test(normdata_nortest)
##
##  Anderson-Darling normality test
##
## data:  normdata_nortest
## A = 0.38284, p-value = 0.3912
ad.test(1:1000)
##
##  Anderson-Darling normality test
##
## data:  1:1000
## A = 11.085, p-value < 2.2e-16

# Smaller p-value indicates significant deviation from normal
# distribution.
```

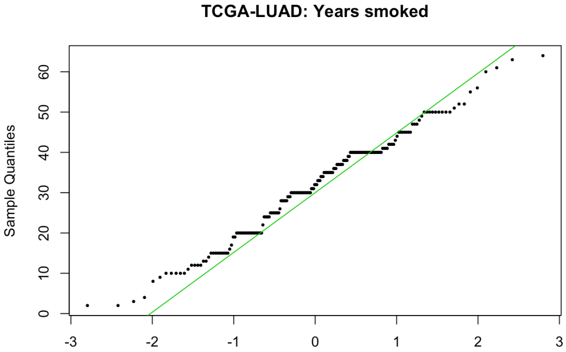
Shapiro-Wilk test of TCGA-LUAD cigarettes per day

```
shapiro.test(tcga_luad$cigarettes_per_day)
##
##  Shapiro-Wilk normality test
##
## data:  tcga_luad$cigarettes_per_day
## W = 0.90998, p-value = 9.873e-14
```



Shapiro-Wilk test of TCGA-LUAD years smoked

```
shapiro.test(tcga_luad$years_smoked)
##
##  Shapiro-Wilk normality test
##
## data:  tcga_luad$years_smoked
## W = 0.98572, p-value = 0.04686
```



One-sample tests of proportion

One-sample tests of proportion

- z-test
- χ^2-test using `prop.test`
- One-sided
- Two-sided
- Power analysis

Overview of one-sample tests of proportion

- Proportion:

$$\frac{\text{The number of observations with certain condition}}{\text{The number of total observations}}$$

- Qusetion: How likely the observed proportion ($prop_{obs}$) is different from another proportion ($prop_{H_0}$)?
  - $prop_{obs}$ is determined by observations.
  - $prop_{H_0}$ is determined by us when specifying the null hypothesis (H_0).

z-test

$$z = \frac{\hat{p} - p_0}{\sqrt{\frac{p_0(1-p_0)}{n}}}$$

Intuition: apply central limit theorem on *n* random variales *i.i.d*. Bernoulli distribution.

Theorem

Let $X_1, X_2, ..., X_n$ be a sequence of i.i.d. random variables with mean $E[X_i] = \mu$ and finite variance $Var(X_i) = \sigma^2$. Define $S_n = \frac{1}{n}\sum_i X_i$. Then, as $n \to \infty$, $S_n \xrightarrow{\mathcal{D}} \mathcal{N}\left(\mu, \sigma^2/n\right)$.

z-test

Example: A new study is performed where 22 out of 200 patients smoked more than 5 cigarettes per day. We would like to check if this group of patients has a statistically greater proportion of heavy smokers compared to the TCGA study. In TCGA, the porportion of heavy smokers 0.1.

Null hypothesis: The porportion of heavy smokers in this new study is no more than the porportion of heavy smokers in the TCGA study.

z-test

Null hypothesis: The porportion of heavy smokers in this new study is no more than 0.1.

```
alpha <- 0.05
z0 <- qnorm(1 - alpha)
print(z0)
## [1] 1.644854

prop2 <- 22/200
p0 <- 0.1
n <- 200

z <- (prop2 - p0)/sqrt(p0 * (1 - p0)/n)
print(z)
## [1] 0.4714045

z >= z0
## [1] FALSE
```

χ^2-test

Intuition: z-test with multiple catogries. For each category, apply central limit theorem on n random variales $i.i.d$. Bernoulli distribution.

$$X^2 = \sum_{i=1}^{k} \frac{(x_i - np_i)^2}{np_i}$$

There are k total categories and n total observations. x_i is the number of observations of category i. p_i is the hypothesized probability of observing category i.

When null hypothesis is correct, as $n \to \infty$, X^2 followed the χ^2 distribution with $(k - 1)$ degrees of freedom stated by CLT.

χ^2-test

Null hypothesis: The porportion of heavy smokers in the new study is no more than 0.1.

```r
alpha = 0.05
z0 = qnorm(1 - alpha)
print(z0)
## [1] 1.644854
hs = 22
p0 = 0.1
n2 = 200

propresult <- prop.test(hs, n2, p = p0, correct = FALSE,
                        alternative = "greater")
```

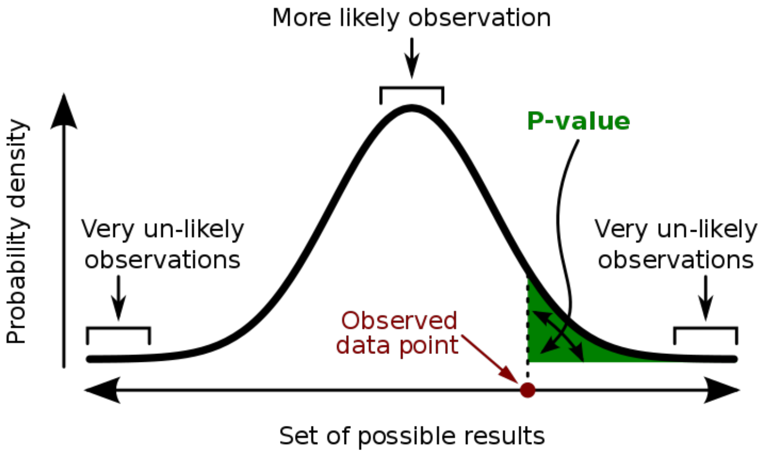
χ^2-test

```
print(propresult)
##
##  1-sample proportions test without continuity correction
##
## data:  hs out of n2, null probability p0
## X-squared = 0.22222, df = 1, p-value = 0.3187
## alternative hypothesis: true p is greater than 0.1
## 95 percent confidence interval:
##  0.0786844 1.0000000
## sample estimates:
##    p
## 0.11
```

χ^2-test

```r
# Comparing prop.test to z-score method.
Xstat <- propresult$statistic[[1]]
print(sqrt(Xstat))
## [1] 0.4714045

z = (hs/n2 - p0)/sqrt(p0 * (1 - p0)/n2)
print(abs(z))
## [1] 0.4714045
```

One-sided versus two-sided test



A **p-value** (shaded green area) is the probability of an observed (or more extreme) result assuming that the null hypothesis is true.

Source: `https://en.wikipedia.org/wiki/P-value`

One-sided versus two-sided test

- Calculate p-value
  - $\Pr(X \geq x|H)$ for right tail event
  - $\Pr(X \leq x|H)$ for left tail event
  - $2min\{\Pr(X \geq x|H), \Pr(X \leq x|H)\}$ for double tail event
- Interpret p-value
  - Important :

$$\Pr(evidence|hypothesis) \neq \Pr(hypothesis|evidence)$$

- Bayes' theorem applied to inference:

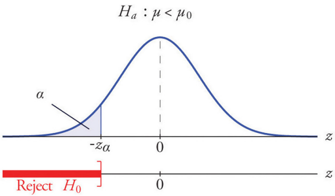
$$P(H|E) = P(E|H) \cdot \frac{P(H)}{P(E)}$$

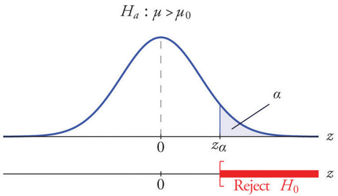
Source: `https://en.wikipedia.org/wiki/P-value`

Source: `https://en.wikipedia.org/wiki/Bayesian_inference`

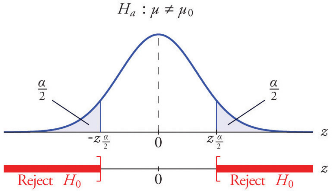
One-sided versus two-sided test

One-sided:





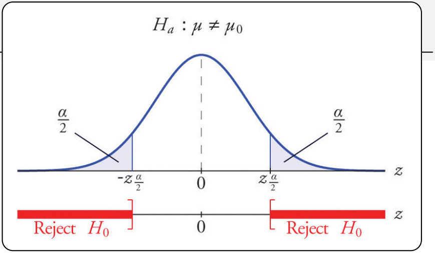
Two-sided:



Source: flatworldknowledge

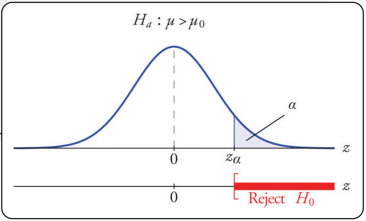
Two-sided one sample test of proportion

```r
hs = 30  # second study patient proportion
p0 = 0.1  # TCGA LUAD proportion
n2 = 200  # second study sample size
z_twotail = (hs/n2 - p0)/sqrt(p0 * (1 - p0)/n)
pval_z_twotail = 2 * pnorm(z_twotail, lower.tail = FALSE)
pval_z_twotail
## [1] 0.01842213

proptest_twotail <- prop.test(hs, n2, p = p0, correct = FALSE,
                              alternative = "two.sided")
proptest_twotail$p.value
## [1] 0.01842213
```



One-sided one sample test of proportion

```
hs = 22  # second study heavy smokers
p0 = 0.1  # TCGA LUAD proportion
n2 = 200  #Number of tests

propresult <- prop.test(hs, n2, p = p0, correct = FALSE,
                        alternative = "greater")
propresult$p.value
## [1] 0.3186759

z = (hs/n2 - p0)/sqrt(p0 * (1 - p0)/n2)
print(abs(z))
## [1] 0.4714045
pvalue_z_onesided <- pnorm(-abs(z))
pvalue_z_onesided
## [1] 0.3186759
```



Power analysis for one-sample test of proportion

- The power of a binary hypothesis test is the probability that the test correctly rejects the null hypothesis (H_0) when a specific alternative hypothesis (H_1) is true.

$$\text{power} = \mathbf{Pr}\left(\text{reject } H_0 \mid H_1 \text{ is true}\right)$$

- Main factors in calculating the power of a test:
  - Effect size, or the difference in means between two groups
  - Sample size: n
  - Significance level, α (often 0.05)

Source: `https://en.wikipedia.org/wiki/Statistical_power`

Power analysis for one-sample test of proportion

Calculate power given effect size, sample size, and significance level.

```
# h is effect size
pwr.p.test(h = 0.5, n = 55, sig.level = 0.05)
##
##      proportion power calculation for binomial distribution
##      (arcsine transformation)
##
##               h = 0.5
##               n = 55
##       sig.level = 0.05
##           power = 0.9597797
##     alternative = two.sided
```

Power analysis for one-sample test of proportion

Calculate sample size given power, effect size, and significance level.

```
pwr.p.test(h = 0.5, n = NULL, sig.level = 0.05, power = 0.9)
##
##      proportion power calculation for binomial distribution
##      (arcsine transformation)
##
##               h = 0.5
##               n = 42.02968
##       sig.level = 0.05
##           power = 0.9
##     alternative = two.sided
```

Power analysis for one-sample test of proportion

Calculate effect size given power, sample size, and significance level.

```
pwr.p.test(h = NULL, n = 100, sig.level = 0.05, power = 0.9)
##
##      proportion power calculation for binomial distribution
##      (arcsine transformation)
##
##               h = 0.3241514
##               n = 100
##       sig.level = 0.05
##           power = 0.9
##     alternative = two.sided
```

Power analysis for one-sample test of proportion

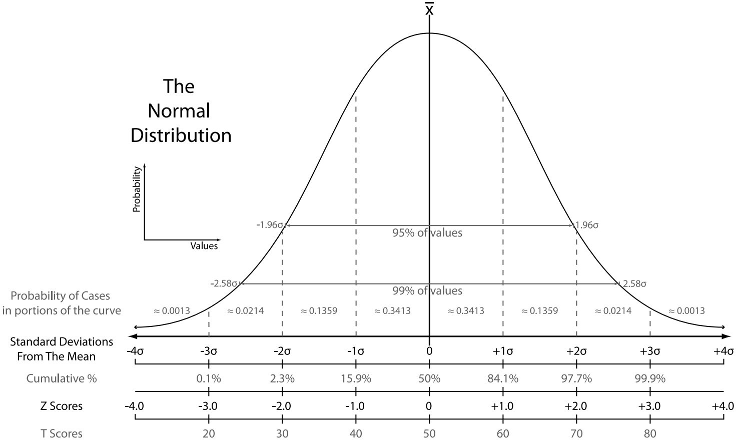
Estimate effect size `h` using `ES.h`.

```
pwr.p.test(h = ES.h(p1 = 0.75, p2 = 0.5), sig.level = 0.05,
           n = NULL, power = 0.8,
           alternative = "greater")
##
##      proportion power calculation for binomial distribution
##      (arcsine transformation)
##
##               h = 0.5235988
##               n = 22.55126
##       sig.level = 0.05
##           power = 0.8
##     alternative = greater
```

Questions?

Appendix

z-score


The Normal Distribution
x̄
Probability
Values
-1.96σ
1.96σ
95% of values
-2.58σ
2.58σ
99% of values
Probability of Cases in portions of the curve
≈ 0.0013
≈ 0.0214
≈ 0.1359
≈ 0.3413
≈ 0.3413
≈ 0.1359
≈ 0.0214
≈ 0.0013
Standard Deviations From The Mean
-4σ
-3σ
-2σ
-1σ
0
+1σ
+2σ
+3σ
+4σ
Cumulative %
0.1%
2.3%
15.9%
50%
84.1%
97.7%
99.9%
Z Scores
-4.0
-3.0
-2.0
-1.0
0
+1.0
+2.0
+3.0
+4.0
T Scores
20
30
40
50
60
70
80