Practical 2 - Estimation, hypothesis testing, and power

Sarah Lewington & Diego Aguilar-Ramirez

2022-05-12

Contents

DISCLAIMER This teaching material is still in development. Please contact the author to report mistakes and for general feedback at diego.aguilar-ramirez@ndph.ox.ac.uk

1. Set-up

Basic descriptive statistics of the Whitehall Resurvey Study baseline data (1997)

In this section you will get to know the Whitehall data by calculating some basic descriptive statistics. This will require you to revisit some of the commands you learned in the **Pre-course practical** and in **Practical 1**.

The Whitehall Resurvey Study is a cohort follow-on study to the original Whitehall study set up in the 1960s, which followed London based male civil servants with a view to investigating cardiovascular disease and mortality. This resurvey in 1997 was conducted to assess the predictive value of blood pressure and blood lipids in old age. It contains information on 4,329 individuals, and the variables are summarised in the table below.

Table 1: List of variables in Whitehall data set

Name	Description
WHL1_D	Participant ID in this dataset
AGE	Age
HATTACK	Prior disease: heart attack diagnosed
ANGINA	Prior disease: angina diagnosed
STROKE	Prior disease: stroke diagnosed
DIAB	Prior disease: diabetes diagnosed

Name	Description
CANCER	Prior disease: cancer diagnosed
SBP	Systolic blood pressure
BMI	Body-mass index (kg/m^2)
CURRSMOKER	Current smoker (1=yes, 0=no)
HDLC	HDL cholesterol (mmol/L)
LDLC	LDL cholesterol (mmol/L)
APOB	ApoB (g/L)
APOA1	ApoA1 (g/L)
CHOL	Total cholesterol (mmol/L)
CPR	C-reactive protein (mmol/l)
VitD	Vitamin D [25(OH)D] nmol/L

1.1 Load the dataset

Be aware of where you have downloaded and saved your file to. If you are unsure of the path, find the file, right click, hit 'Properties' on the menu, and then copy the file path under the heading 'Location'. Ideally, you will have saved your data set within your **R Project folder** (see Pre-course material).

Open R and open the R Project you have been using for the course.

Let's import the Whitehall data set as we have done before.

First, let's load the packages we will use:

```
pacman::p_load(rio,here,skimr,ggplot2) #Load the packages with pacman
```

Use **here** to check your current working directory, which should be the directory of the R Project

```
here()
```

If everything looks okay, let's import the data. If you have opened your R Project and the `Whitehall_like-Baseline.csv` file is saved in the `~/my_project/data` subfolder, then this code should work:

```
w.hall.data <- import(here("data","Whitehall_like-Baseline.csv"), na.strings = ".")
```

Recall that **missing data** in R appears as `NA`. `NA` is not a string or a numeric value, but an indicator of missingness. You can use the argument `na.strings` to tell `import()` which strings should be interpreted as `NA`. In the Whitehall dataset, the missing indicator for continuous variables is `.`. If the argument `na.strings` is omitted, the missing values `.` will be interpreted as `characters` and then the whole column will be converted to characters.

1.2 Explore the data

Before doing any analyses you should always examine your data to get a feel for the data.

Hint: use commands from previous session > `summary()`, `dim()`, `nrow()`, `ncol()`, `names()`, `skim()`.

Let's try using `skim()` from the **skimr** package:

```
skim(w.hall.data)
```

Table 2: Data summary

Name	w.hall.data
Number of rows	4327
Number of columns	17

Table 2: Data summary

Column type frequency:	
numeric	17
Group variables	None

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
WHL1_ID	0	1.00	12164.00	1249.24	10001.00	11082.50	12164.00	13245.50	14327.00	
AGE	0	1.00	76.83	4.86	58.00	73.00	76.00	80.00	96.00	
HATTACK	0	1.00	0.11	0.32	0.00	0.00	0.00	0.00	1.00	
ANGINA	0	1.00	0.14	0.35	0.00	0.00	0.00	0.00	1.00	
STROKE	0	1.00	0.07	0.26	0.00	0.00	0.00	0.00	1.00	
DIAB	0	1.00	0.06	0.24	0.00	0.00	0.00	0.00	1.00	
CANCER	0	1.00	0.08	0.26	0.00	0.00	0.00	0.00	1.00	
SBP	9	1.00	130.75	17.57	86.00	119.00	129.00	141.00	230.00	
BMI	17	1.00	25.22	3.26	15.00	23.00	25.00	27.00	44.00	
CURRSMOKER	6	1.00	0.13	0.33	0.00	0.00	0.00	0.00	1.00	
HDLC	25	0.99	1.09	0.38	0.05	0.82	1.05	1.32	3.07	
LDLC	25	0.99	3.37	0.78	1.05	2.84	3.35	3.87	6.81	
APOB	5	1.00	0.87	0.23	0.24	0.71	0.84	1.00	2.42	
APOA1	5	1.00	0.95	0.14	0.52	0.85	0.93	1.03	2.23	
CHOL	5	1.00	5.51	1.01	2.24	4.85	5.47	6.15	10.77	
CRP	25	0.99	3.61	7.50	0.12	0.86	1.71	3.58	210.00	
VitD	0	1.00	57.35	17.98	18.92	45.60	55.70	66.69	419.89	

Remember: `skim()` is a simple way to quickly explore your data but it is not the only way. Moreover, `summary()`, `dim()`, `nrow()`, `ncol()`, `names()`, and other functions from **Basic R** are really helpful for running quick checks while coding and are powerful tools for analyses. Learning different ways of doing the same think might feel unnecessary, but it is actually quite handy.

---

2. Calculating standard errors and confidence intervals manually

Here you will calculate SE and confidence intervals by hand, and interpret confidence intervals.

2.1 Calculating SE

In the lecture you saw that:

$$SE = \frac{SD}{\sqrt{n}}$$

where SD is the population standard deviation, n is your sample size, and SE is the standard deviation of the sampling distribution.

Question 2.1.1 Fill in the table bellow, with answers to 3 decimal places:

	APOB	APOA1	CRP	LDLC	VitD
n					
SD					
SE					

HINT: use the following commands

To calculate n

```
APOB.n <- sum(!is.na(w.hall.data$APOB)) #remember to ask R not to include missing data
APOB.n
```

To calculate SD

```
APOB.sd <- sd(w.hall.data$APOB,na.rm=T) #remember to ask R to remove missing valyes
APOB.sd
round(APOB.sd,digits=3)
```

To calculate SE

```
APOB.se <- APOB.sd / sqrt(APOB.n)
APOB.se
round(APOB.se,3)
```

Results to question 2.1.1

The resulting table should look like this:

Table 5: Calculating SD and SE

	APOB	APOA1	CRP	LDLC	VitD
n	4322.000	4322.000	4302.000	4302.000	4327.000
SD	0.234	0.145	7.499	0.780	17.976
SE	0.004	0.002	0.114	0.012	0.273

2.2 Calculating confidence intervals

A 95% confidence interval for the sample mean is calculated using the formula:

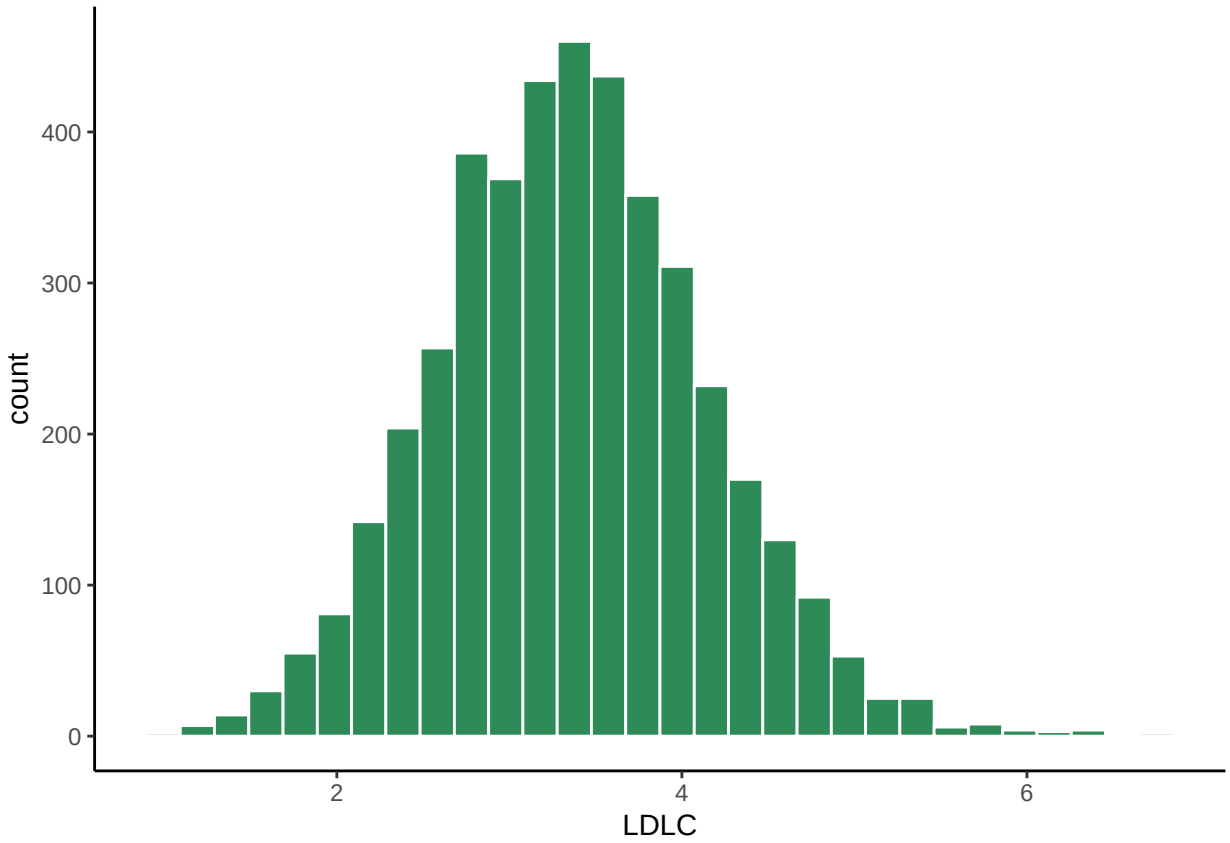
$$[\hat{x} - Z_{0.975} * SE, \hat{x} + Z_{0.975} * SE]$$

NOTE: $Z_{0.975}$ is the 97.5% percentile of the standard normal distribution (~1.96 SD from the mean). This can be determined more precisely using the R function `qnorm(0.975)`.

Question 2.2.1 Plot a histogram for the variable LDLC and determine if this variable is approximately normally distributed.

HINT: Use code from previous sessions

```
ggplot(w.hall.data, aes(LDLC)) + # Plot histogram
  geom_histogram(colour="white", fill="seagreen") + # A little bit of colour
  theme_classic() # Use a ggplot classic theme
```



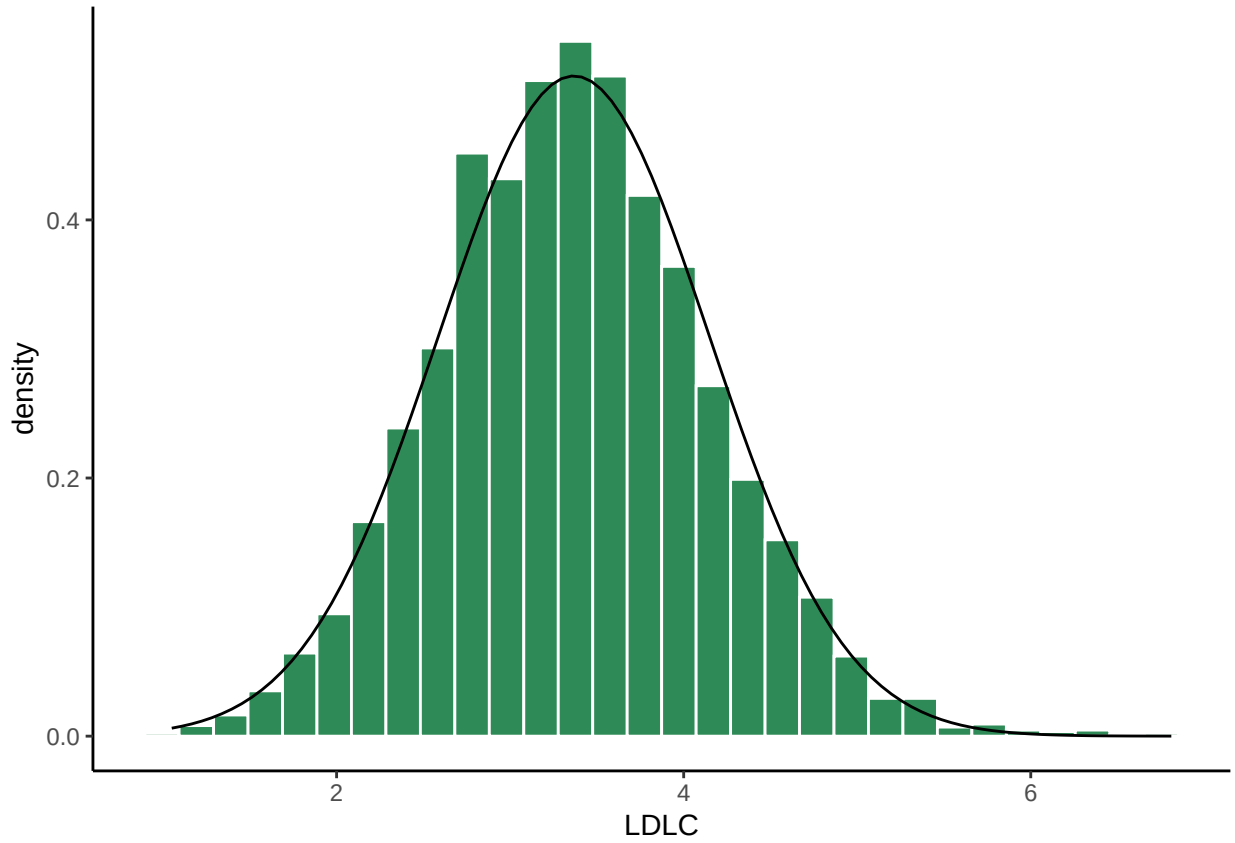
Result to question 2.2.1

It does look like the sample distribution of LDLC is approximately normally distributed.

Let's compare it with a normal curve:

```
# For this we will need a bit more code
LDLC.mu = mean(w.hall.data$LDLC,na.rm = TRUE ) #Define sample mean
LDLC.sd = sd(w.hall.data$LDLC,na.rm = TRUE ) # Define SD

ggplot(w.hall.data, aes(LDLC)) + # Plot "histogram"
  geom_histogram(aes(y=..density..), # Using density instead of counts as y-axis
                 colour="white", fill="seagreen") + # A little bit of colour
  stat_function(fun = dnorm, args = list(mean = LDLC.mu, sd = LDLC.sd)) + #Add normal distribution curv
  theme_classic()
```



Question 2.2.2 Using the formula above, calculate a 95% confidence interval for the mean LDL.

You can use a calculator (using ~1.96) and the data of the table above.

Or you can use the following code:

```
LDLC.n    <- sum(!is.na(w.hall.data$LDLC)) # Sample size
Za        <- qnorm(0.975) # Normal distribution cut-off
ci.Z_LDLC <- c(
  LDLC.mu-(LDLC.sd/sqrt(LDLC.n)*Za), # Lower limit of CI
  LDLC.mu+(LDLC.sd/sqrt(LDLC.n)*Za)  # Upper limit of CI
  )
```

Notice how `LDLC.sd/sqrt(LDLC.n)` is effectively $\frac{SD}{\sqrt{n}}$ which is the SE. The implication of this is that the larger n is (i.e., the sample size), the smaller SE and the narrower the CI will be (i.e., there will be less uncertainty around the estimated mean).

Result to question 2.2.2

```
ci.Z_LDLC # Display in console
```

```
## [1] 3.344065 3.390672
```

One way of interpreting this interval is as follows: "*We are 95% confident that the true mean LDL cholesterol (for this population) lies between 3.344 and 3.390 mmol/L*"

3. Two sample t-test

In this section you will perform a t-test and interpret the results

NOTE: the t-test is based on Student's t-distribution, which is slightly different when the sample size is small (n<30).

In larger sample sizes (n>=30), the t-distribution is effectively the same as the z-distribution (Gaussian or normal distribution, which was discussed in the lecture).

You can find here more information about how z and t tests compare using R.

3.1 Running a t-test for SBP and age

For a one-sample t-test, the syntax is:

```
t.test( x , conf.level = 0.95 , ...[options] )
```

For a two-sample t-test, the following code should be used:

```
t.test( x[group == 1], x[group == 2] , conf.level = 0.95 , ...[options] )
```

For a two-sample test, x specifies the variable on which to perform the t-test, while the group variable defines the groups to be compared. An alternative formulation for the two sample t.test is:

```
t.test( x ~ group, data = white.data)
```

See `?t.test` for details of the other options. One of the options available accounts for unequal variances across the groups, but you do not need to worry about this now.

Perform a t-test to compare the mean `LDLC` in those aged under 76 years with those aged 76 years old or more.

First, let's recode the variable `AGE` into a binary variable `agegp`. This can be achieved by adding a new variable to the dataframe `w.hall.data`

```
w.hall.data$agegp <- NA
w.hall.data$agegp[w.hall.data$AGE <  76] = 1
w.hall.data$agegp[w.hall.data$AGE >= 76] = 2
```

Now check the recoding worked using `table()`

```
table(w.hall.data$agegp)
```

```
##
##    1    2
## 1837 2490
```

```
t.test(LDLC~agegp, data=w.hall.data)
```

Question 3.1.1: Complete the following table and interpret the results.

T-test comparing `LDLC` in those aged under 76 and aged 76 or more

Description	Value
mean LDLC where age<76 (x1)	
mean LDLC where age>=76 (x2)	
mean difference (x1-x2)	
test statistic t	
95% CI for mean difference	
p-value	

Result to question 3.1.1

```r
t.test(LDLC~agegp, data=w.hall.data)
```

```
##
##  Welch Two Sample t-test
##
## data:  LDLC by agegp
## t = 5.5172, df = 3994.6, p-value = 3.662e-08
## alternative hypothesis: true difference in means between group 1 and group 2 is not equal to 0
## 95 percent confidence interval:
##  0.08489092 0.17848060
## sample estimates:
## mean in group 1 mean in group 2
##        3.443160        3.311474
```

By default the test does not assume that variances of the two groups are the same and tests the null hypothesis against the **'two-sided'** alternative that the difference in means is not equal to zero.

The output shows the test statistic ('t') for the test (mean of the difference/SE of the difference) and the degrees of freedom ('df') for the test. On the same line it also gives the p-value for the two tailed tests that hypothesise the mean difference could be in any direction. This p-value is extremely small (<0.0001) indicating strong evidence against the null hypothesis.

The 95% confidence interval for the difference in the means is also reported along with the means in each age group.

T-test comparing `LDLC` in those aged under 76 and aged 76 or more

Description	Value
mean LDLC where age<76 (x1)	3.44
mean LDLC where age>=76 (x2)	3.31
mean difference (x1-x2)	0.13
test statistic t	5.517
95% CI for mean difference	(0.085-0.178)
p-value	0

Interpret this result:

- There is sufficient evidence, at 5% level, to reject the null hypothesis; OR
- There is statistically significant difference, at 5% level, between the two groups

---

4. Power and sample size

In this section, you will practice some general commands for estimating power and sample sizes, irrespective of study design.

4.1 The `power` function

Calculations for power and sample size in R can be performed using the `power.test` functions. If you look at the help file, you will see that you can use these functions to compute a sample size, the power, or an effect size.

You do not need to have a data set loaded.

Power calculations for two means

The syntax looks like this:

```
power.t.test(n = NULL, delta = NULL, sd = 1, sig.level = 0.05,
             power = NULL,
             type = c("two.sample", "one.sample", "paired"),
             alternative = c("two.sided", "one.sided"))
```

Arguments

n	Number of observations (per group)
delta	True difference in means
sd	Standard deviation
sig.level	Significance level (Type I error probability)
power	Power of test (1 minus Type II error probability)
type	Type of t test
alternative	One- or two-sided test. Can be abbreviated

Exactly one of the parameters `n`, `delta`, `power`, `sd`, and `sig.level` must be passed as `NULL`, and that parameter is determined from the others.

See `help(power.t.test)` for more details.

You need to choose the method you are calculating a power estimate for. To do this, ask yourself if you have 1 sample, 2 independent samples or 2 paired samples? And do you want to compare means from continuous outcomes or proportions from binary/categorical outcomes (see `power.prop.test`)?

Before proceeding, remind yourself of some of the key concepts from the lecture:

Question 4.1.1:

- What information/parameters do you need before you can estimate the sample size needed for a particular study?

Result to question 4.1.1

You need to choose a level of significance you will accept and the amount of power you want to have, i.e. the chance of detecting a difference if one exists. For two means you would need the difference in the means and the standard deviation (SD).

Question 4.1.2

- What information/parameters do you need before you can calculate the power of a given sample size and study?

Result to question 4.1.2

You need to determine a level of significance and the sample size for means. For two means you would need the difference in the means and the standard deviation (SD).

Using the `power.t.test` function, answer the following questions.

Question 4.1.3

Estimate the **sample size** needed to compare the mean systolic blood pressure (SBP) in two populations. From a pilot study, you think that the group with lower blood pressure will have a mean SBP of 120 mm Hg, and the standard deviation (SD) of both groups will be 15 mm Hg. You have decided that you are interested in a minimum difference of 5 mm Hg, and you want 90% power, and a 5% significance level.

Result to question 4.1.3

```
power.t.test(delta=5, sd=15, sig.level=0.05, power=0.9)
```

```
##
##      Two-sample t test power calculation
##
##               n = 190.0991
##           delta = 5
##              sd = 15
##       sig.level = 0.05
##           power = 0.9
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

The estimated sample size: 382 (191 per group)

Question 4.1.4

Estimate the sample size needed to compare the mean total cholesterol between two groups of men, with versus without coronary heart disease (CHD). Based on a previous study, you think the mean total cholesterol will be 5.5 mmol/l (SD: 1 mmol/l) amongst men without CHD, and 6.0 mmol/l (SD: 1 mmol/l) amongst men with CHD. Assume you want 90% power, and a 5% significance level.

Result to question 4.1.4

```
power.t.test(delta=0.5, sd=1, sig.level=0.05, power=0.9)
```

```
##
##      Two-sample t test power calculation
##
##               n = 85.03129
##           delta = 0.5
##              sd = 1
##       sig.level = 0.05
##           power = 0.9
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

The estimated sample size: 172 (86 per group)

Question 4.1.5

You have now found some other studies, looking at the same question as in question **4.1.4**, which give different estimates for the difference between the mean values of total cholesterol in the two groups, varying from 0.1 mmol/l to 1.0 mmol/l. What would be your estimated sample size if the mean total cholesterol in men with CHD is 1.0 mmol/l higher than in men without CHD? What if it was only 0.25 mmol/l higher? [Assume that your previous estimate of mean cholesterol amongst men without CHD (5.5 mmol/l) is correct].

Result to question 4.1.5
Difference of 1.0 mmol/L

```
power.t.test(delta=1, sd=1, sig.level=0.05, power=0.9)
```

```
##
##      Two-sample t test power calculation
##
##               n = 22.0211
##           delta = 1
```

```
##              sd = 1
##       sig.level = 0.05
##           power = 0.9
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

Difference of 0.25 mmol/L

```
power.t.test(delta=0.25, sd=1, sig.level=0.05, power=0.9)
```

```
##
##      Two-sample t test power calculation
##
##               n = 337.2008
##           delta = 0.25
##              sd = 1
##       sig.level = 0.05
##           power = 0.9
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

Note how much the sample size needed changes, depending on the magnitude of the difference we expect to detect.

Question 4.1.6

Plot a graph to illustrate how power changes with sample size (200 to 2000), for the study described in **4.1.3**

Hints:

Use a vector in order to generate a list of powers for a list of sample sizes; we can generate a list of sample sizes using the following command:

```
samplesizes <- seq(from=200, to=2000, by=200)
```

And then use this instead of a value in our `power.prop.test`.

- Use $power to save only the powers from the results
- Use `geom_point()`
- Add `gep,_line` to connect dots

Result to question 4.1.6
Define the sample sizes and calculate the power for each sample size

```
samplesizes <- seq(from=200, to=2000, by=200)
power.samplesizes <- power.prop.test(n=samplesizes, p1=0.1, p2=0.15)$power
```

Have a look at the calculated powers

```
power.samplesizes
```

```
##  [1] 0.3265745 0.5708887 0.7455465 0.8569699 0.9228823 0.9597956 0.9796164
##  [8] 0.9899067 0.9951027 0.9976656
```

Now, let's plot our results

```
ggplot(mapping=aes(x=samplesizes, y=power.samplesizes)) +
  geom_line() +
  geom_point() +
  theme_classic() +
```

```
  xlab("Sample size") +
  ylab("Expected power") +
  ggtitle("Power by sample size")
```