Assignment 08

Zhen Liu

2022-11-28

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──
## ✔ ggplot2 3.3.6      ✔ purrr   0.3.4
## ✔ tibble  3.1.8      ✔ dplyr   1.0.10
## ✔ tidyr   1.2.1      ✔ stringr 1.4.1
## ✔ readr   2.1.3      ✔ forcats 0.5.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

Obstacles to valid scientific inference

Q1

1. Measurement distortions

your data has errors

Measurement error is very common in practice - Miscalibration of measurement instruments e.g. a clock running slightly too fast or slow. - Rounding errors due to computational constraints e.g. 0.904 converted to 0.9. - Inaccurate responses to questionnaires e.g. under reporting alcohol consumption. - Human error e.g. data entry mistakes

2. Selection bias

Selection bias occurs when the data included in the analysis misrepresents the underlying population of interest. sample bias, self-selection bias, attrition bias, and post-hoc selection

sample bias: You want to know what the most popular genre (jazz, folk, rock …) is amongst attendees of a music festival

self selection bias: online reviews of restaurants might disproportionately represent subsets of the population with strong opinions or certain age groups

attrition bias:A scientist is investigating the efficacy of a new exercise program

post hoc selection: A scientist is investigating the efficacy of medical treatment

Randomized samples: sampling from the population of interest which is the set of all attendees of a music festival

3. Confounding variables

Correlation does not imply causation

Example: the increased sales of sun glass may not be the reason for the increased sales of ice cream, but the sunny weather is

paired t-test and effect size

```
library(PairedData)
```

```
## Loading required package: MASS
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
## Loading required package: gld
```

```
## Loading required package: mvtnorm
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'PairedData'
```

```
## The following object is masked from 'package:base':
##
##     summary
```

```
data("Barley")
detach('package:PairedData',unload = TRUE)
detach('package:MASS', unload = TRUE)

head(Barley,4)
```

```
##   Farm Glabron Velvet
## 1  F01      49     42
## 2  F02      47     47
## 3  F03      39     38
## 4  F04      37     32
```

Q1

```
t.test(x =Barley$Glabron,y =Barley$Velvet,paired = TRUE )
```

```
##
##  Paired t-test
##
## data:  Barley$Glabron and Barley$Velvet
## t = 3.0133, df = 11, p-value = 0.0118
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
##   1.594978 10.238355
## sample estimates:
## mean difference
##        5.916667
```

The p-value is 0.0118 which is above the significant level of 0.01. Thus we do not reject the null hypothesis

Q2

```
y_bar<-mean(Barley$Glabron-Barley$Velvet)
s<- sd(Barley$Glabron-Barley$Velvet)
effect_size<-y_bar/s
effect_size
```

```
## [1] 0.8698615
```

Q3

1. Subjects must be independent. Measurements for one subject do not affect measurements for any other subject.
2. Each of the paired measurements must be obtained from the same subject. For example, the before-and-after weight for a smoker in the example above must be from the same person.
3. The measured differences are normally distributed

Implementing unpaired t-test

```
library(palmerpenguins)
peng_AC<-penguins %>%
  drop_na(species,body_mass_g) %>%
  filter(species !="Gentoo")
head(peng_AC %>%
       select(species, flipper_length_mm, body_mass_g), 5)
```

```
## # A tibble: 5 × 3
##   species flipper_length_mm body_mass_g
##   <fct>               <int>       <int>
## 1 Adelie                181        3750
## 2 Adelie                186        3800
## 3 Adelie                195        3250
## 4 Adelie                193        3450
## 5 Adelie                190        3650
```

Q1

```
val_col <- "body_mass_g"
group_col <- "species"
data <- peng_AC
data_new <- data %>%
  # rename the columns; note that you can not drop the "!!" (why?)
  rename(group=(!!group_col),val=(!!val_col))%>%
  group_by(group) %>%
  drop_na(val) %>%
  summarise(mn =val)
```

```
## `summarise()` has grouped output by 'group'. You can override using the
## `.groups` argument.
```

```
data_new
```

```
## # A tibble: 219 × 2
## # Groups:   group [2]
##    group     mn
##    <fct>  <int>
##  1 Adelie  3750
##  2 Adelie  3800
##  3 Adelie  3250
##  4 Adelie  3450
##  5 Adelie  3650
##  6 Adelie  3625
##  7 Adelie  4675
##  8 Adelie  3475
##  9 Adelie  4250
## 10 Adelie  3300
## # … with 209 more rows
```

```
data_new$mn[2]
```

```
## [1] 3800
```

```
t_test_function<-function(data=peng_AC,group_col="body_mass_g",val_col = "species"){

val_col <- "body_mass_g"
group_col <- "species"
data <- peng_AC

data_new <- data %>%
    # rename the columns; note that you can not drop the "!!" (why?)
    rename(group=(!!group_col),val=(!!val_col))%>%
    group_by(group) %>%
    drop_na(val) %>%
    summarise(mn=mean(val),sd = sd(val),n=length(val))

sd_combined<-sqrt(((data_new$n[1]-1)*data_new$sd[1]^2 + (data_new$n[2]-1)*data_new$sd
[2]^2)/(data_new$n[1]+data_new$n[2]))

t_statistic<- (data_new$mn[1]-data_new$mn[2])/(sd_combined*sqrt(1/data_new$n[1]+1/dat
a_new$n[2]))


p_value<-2*(1-pt(abs(t_statistic),df = data_new$n[1]+data_new$n[2]))

effect_size<- (data_new$mn[1]-data_new$mn[2])/(data_new$sd[1]-data_new$sd[2])

output_df<- data.frame(t_statistic,p_value,effect_size)


return(output_df)
}
```

```
t_test_function(data=peng_AC,group_col="body_mass_g",val_col = "species")
```

```
##   t_statistic   p_value effect_size
## 1   -0.510423 0.6102692  -0.4368251
```

Q2

```
t_test_function<-function(data=peng_AC,group_col="body_mass_g",val_col = "species",va
l_equal=FALSE){

val_col <- "body_mass_g"
group_col <- "species"
data <- peng_AC

data_new <- data %>%
    # rename the columns; note that you can not drop the "!!" (why?)
    rename(group=(!!group_col),val=(!!val_col))%>%
    group_by(group) %>%
    drop_na(val) %>%
    summarise(mn=mean(val),sd = sd(val),n=length(val))


welch_test<-t.test(data_new$group~data_new$mn, data = data_new)


return(welch_test)


}
```

```
# cannot figure out
#t_test_function(data=peng_AC,val_col="body_mass_g",group_col="species", val_equal=FA
LSE)
```

Useful concepts in statistical hypothesis testing

Q1

1. Null hypothesis

A null hypothesis is a type of statistical hypothesis that proposes that no statistical significance exists in a set of given observations.

2. Alternative hypothesis

The alternative hypothesis (HA) is the other answer to your research question. It claims that there's an effect in the population.

Your alternative hypothesis is the same as your research hypothesis. In other words, it's the claim that you expect or hope will be true.

3. Test statistic

A test statistic is a statistic (a quantity derived from the sample) used in statistical hypothesis testing. A hypothesis test is typically specified in terms of a test statistic, considered as a numerical summary of a data-set that reduces the data to one value that can be used to perform the hypothesis test.

4. Type 1 error

A Type I error means rejecting the null hypothesis when it's actually true. It means concluding that results are statistically significant when, in reality, they came about purely by chance or because of unrelated factors.

5. Type 2 error

A Type II error means not rejecting the null hypothesis when it's actually false. This is not quite the same as "accepting" the null hypothesis, because hypothesis testing can only tell you whether to reject the null hypothesis.

Instead, a Type II error means failing to conclude there was an effect when there actually was. In reality, your study may not have had enough statistical power to detect an effect of a certain size.

6. The size of a test

In statistics, the size of a test is the probability of falsely rejecting the null hypothesis. That is, it is the probability of making a type I error. It is denoted by the Greek letter α (alpha).

7. The power of a test

The power of a hypothesis test is the probability that the test correctly rejects the null hypothesis. That is, the probability of a true positive result. It is only useful when the null hypothesis is rejected.

8. The significance level

The significance level is the probability of rejecting the null hypothesis when it is true. For example, a significance level of 0.05 indicates a 5% risk of concluding that a difference exists when there is no actual difference.

9. The p-value

The P value is defined as the probability under the assumption of no effect or no difference (null hypothesis), of obtaining a result equal to or more extreme than what was actually observed

10. Effect size

Effect size tells you how meaningful the relationship between variables or the difference between groups is. It indicates the practical significance of a research outcome. A large effect size means that a research finding has practical significance, while a small effect size indicates limited practical applications.

Q2

False

1. It needs consider the assumption of no effect or no difference (null hypothesis), of obtaining a result equal to or more extreme than what was actually observed
2. No . Statistical significance doesn't tell us about magnitude of difference

Investigating test size for an unpaired student's t-test

```
num_trials<-10000
sample_size<-30
mu_0<-1
mu_1<-1
sigma_0<-3
sigma_1<-3
alpha<-0.05
set.seed(0)


single_alpha_test_size_simulation_df <- data.frame(trial=seq(num_trials)) %>%
  # generate random Gaussian samples
  # generate p values # type I error
  mutate(sample_0=map(.x=trial,.f=~rnorm(n=sample_size,mean=mu_0,sd=sigma_0)), sample
_1=map(.x=trial,.f=~rnorm(n=sample_size,mean=mu_1,sd=sigma_1))) %>%
  mutate(p_value=pmap(.l=list(trial,sample_0,sample_1), .f=~t.test(..2,..3,var.equal
 = TRUE)$p.value))%>%
  mutate(type_1_error=p_value<alpha)

single_alpha_test_size_simulation_df %>%
  pull(type_1_error) %>%
  mean() # estimate of coverage probabilit
```

```
## [1] 0.0502
```

Q1

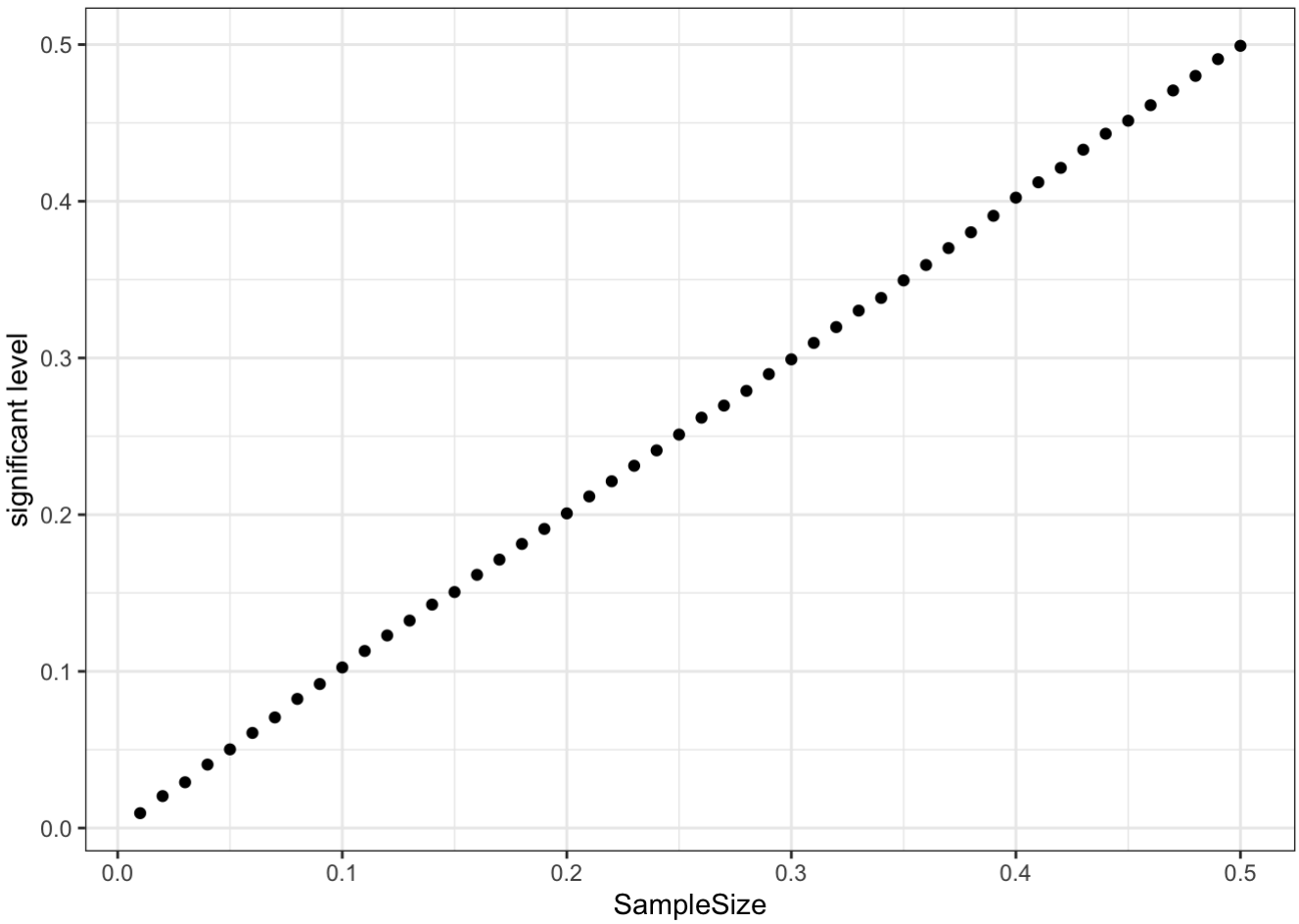
```
significant_a<-function(a){

num_trials<-10000
sample_size<-30
mu_0<-1
mu_1<-1
sigma_0<-3
sigma_1<-3
alpha<-a
set.seed(0)

single_alpha_test_size_simulation_df <- data.frame(trial=seq(num_trials)) %>%
  # generate random Gaussian samples
  # generate p values # type I error
  mutate(sample_0=map(.x=trial,.f=~rnorm(n=sample_size,mean=mu_0,sd=sigma_0)), sample
_1=map(.x=trial,.f=~rnorm(n=sample_size,mean=mu_1,sd=sigma_1))) %>%
  mutate(p_value=pmap(.l=list(trial,sample_0,sample_1), .f=~t.test(..2,..3,var.equal
 = TRUE)$p.value))%>%
  mutate(type_1_error=p_value<alpha)

size<-single_alpha_test_size_simulation_df %>%
  pull(type_1_error) %>%
  mean() # estimate of coverage probabilit

return(size)

}
```

```
significant_a(0.01)
```

```
## [1] 0.0095
```

```
sample_size<-seq(0.01, 0.5, 0.01)

df_size<-data.frame(sample_size)
df_importance<-df_size%>%
  mutate(importance_a=map(sample_size, ~significant_a(.x)) )
```

```
df_importance$importance_a=as.numeric(df_importance$importance_a)

ggplot(df_importance,aes(x=sample_size,y=importance_a))+geom_point()+ theme_bw()+xlab
('SampleSize')+ylab('significant level')
```



The statistical power of an unpaired t-test

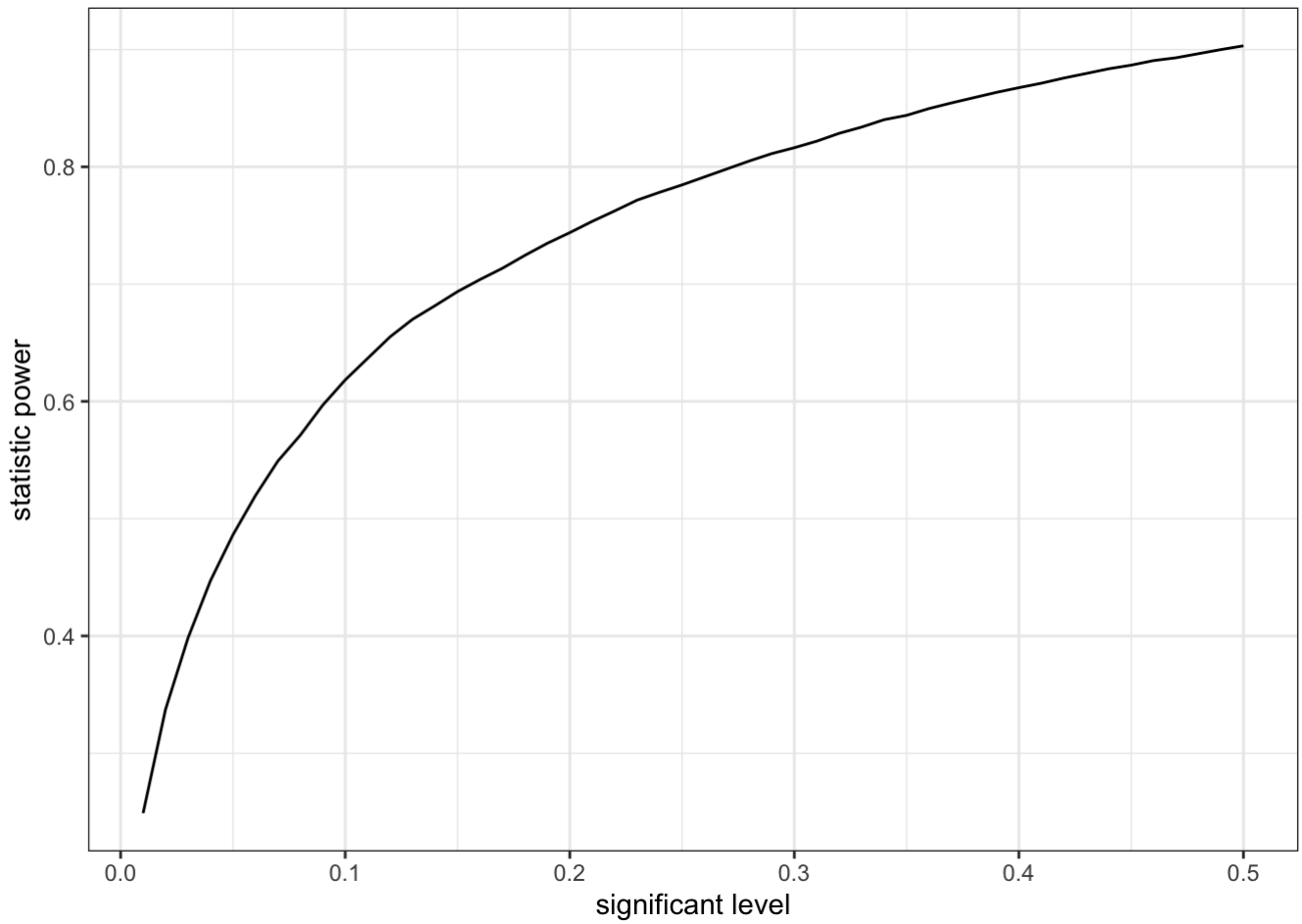
```
num_trials<-10000
n_0<-30
n_1<-30
mu_0<-3
mu_1<-4
sigma_0<-2
sigma_1<-2
alpha<-0.05
set.seed(0)


data.frame(trial=seq(num_trials)) %>%
  mutate(sample_0 = map(.x=trial,.f =~ rnorm(n=n_0,mean=mu_0,sd=sigma_0)), sample_1 =
map(.x=trial,.f =~ rnorm(n=n_1,mean=mu_1,sd=sigma_1))) %>%

  mutate(p_value=pmap(.l = list(trial,sample_0,sample_1), .f =~ t.test(..2, ..3, var.
equal = TRUE)$p.value)) %>% mutate(reject_null = p_value<alpha ) %>% pull(reject_nul
l) %>%
  mean()
```

```
## [1] 0.4862
```

Q1

```
significant_level<-function(a){


num_trials<-10000
n_0<-30
n_1<-30
mu_0<-3
mu_1<-4
sigma_0<-2
sigma_1<-2
alpha<-a
set.seed(0)


result<-data.frame(trial=seq(num_trials)) %>%
  mutate(sample_0 = map(.x=trial,.f =~ rnorm(n=n_0,mean=mu_0,sd=sigma_0)), sample_1 =
map(.x=trial,.f =~ rnorm(n=n_1,mean=mu_1,sd=sigma_1))) %>%
  mutate(p_value=pmap(.l = list(trial,sample_0,sample_1), .f =~ t.test(..2, ..3, var.
equal = TRUE)$p.value)) %>% mutate(reject_null = p_value<alpha ) %>%
  pull(reject_null) %>%
  mean()

return(result)


}
```

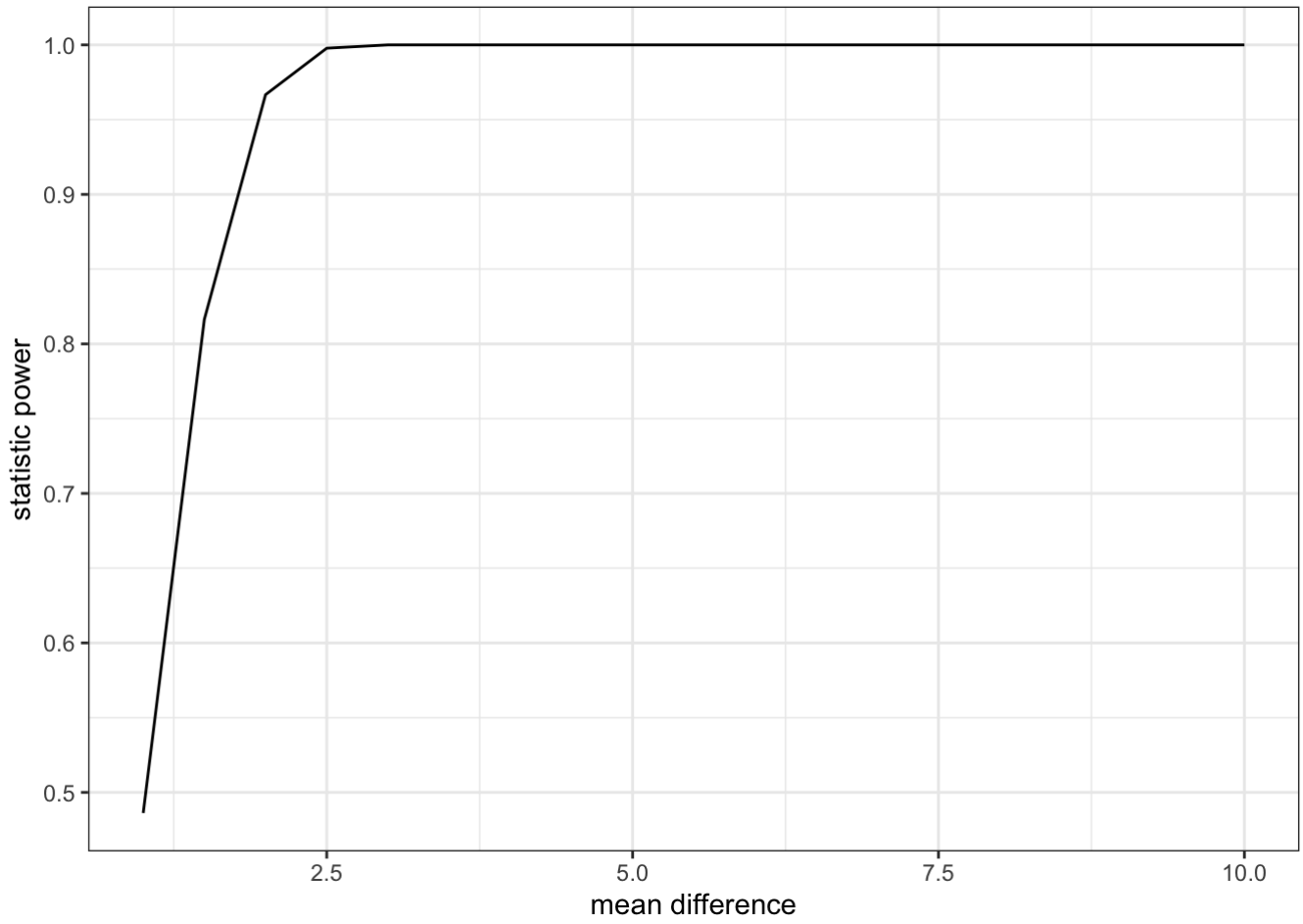
```
sample_size<-seq(0.01, 0.5, 0.01)

df_size<-data.frame(sample_size)
df_importance_level<-df_size%>%
  mutate(importance_a=map(sample_size, ~significant_level(.x)) )


df_importance_level$importance_a=as.numeric(df_importance_level$importance_a)

ggplot(df_importance_level,aes(x=sample_size,y=importance_a))+geom_line()+ theme_bw()
+xlab('significant level')+ylab('statistic power')
```



Q2

```
mean_dff<-function(m0,m1){


num_trials<-10000
n_0<-30
n_1<-30
mu_0<-m0
mu_1<-m1
sigma_0<-2
sigma_1<-2
alpha<-0.05
set.seed(0)


result<-data.frame(trial=seq(num_trials)) %>%
  mutate(sample_0 = map(.x=trial,.f =~ rnorm(n=n_0,mean=mu_0,sd=sigma_0)), sample_1 =
map(.x=trial,.f =~ rnorm(n=n_1,mean=mu_1,sd=sigma_1))) %>%
  mutate(p_value=pmap(.l = list(trial,sample_0,sample_1), .f =~ t.test(..2, ..3, var.
equal = TRUE)$p.value)) %>% mutate(reject_null = p_value<alpha ) %>%
  pull(reject_null) %>%
  mean()

return(result)


}
```

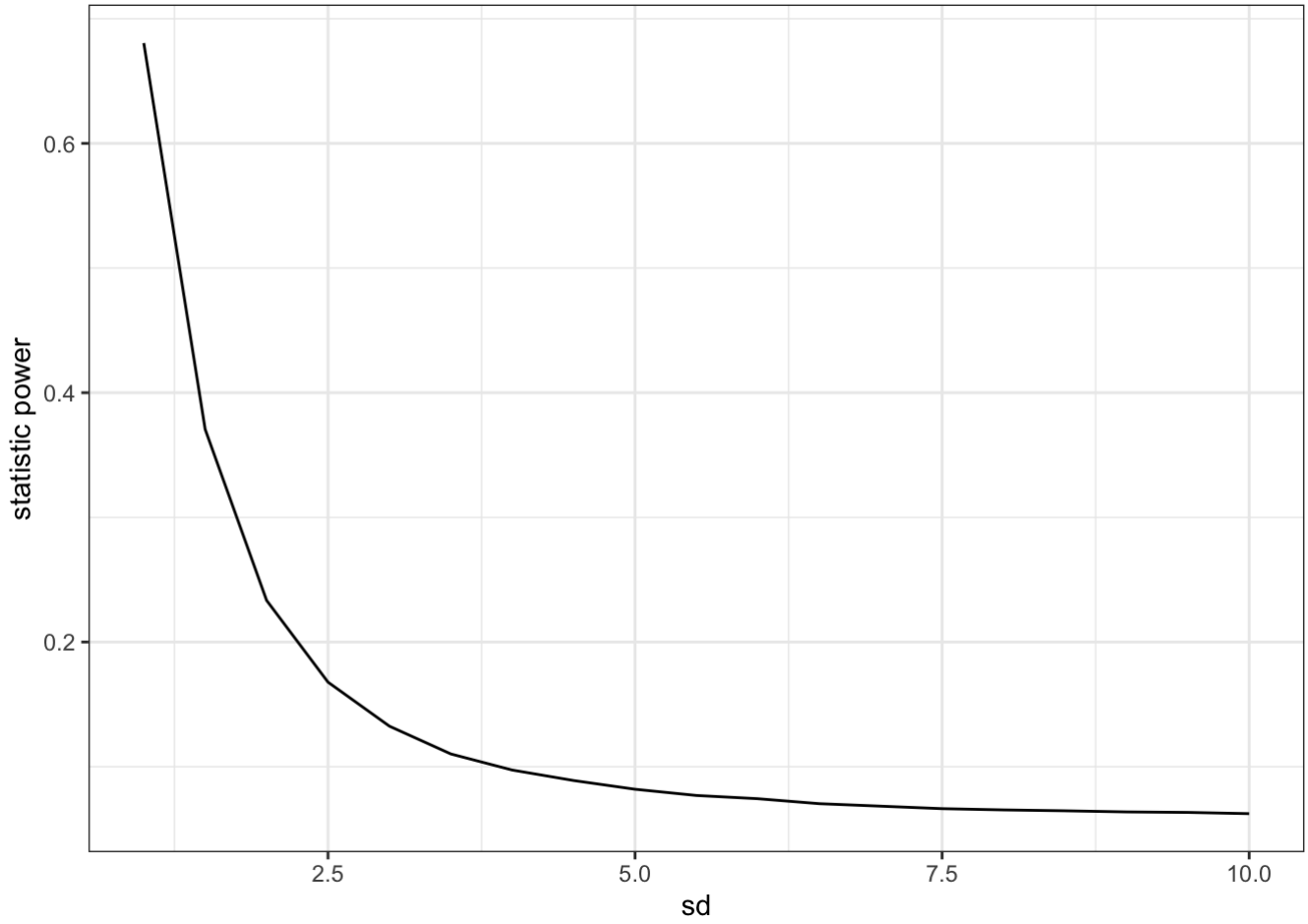
```
m0<-seq(1, 10, 0.5)
diff_m<-seq(1,10,0.5)
m1<-m0+diff_m


df_size<-data.frame(m0,m1,diff_m)
df_mean<-df_size%>%
  mutate(importance_a=map2_dbl(m0,m1, ~mean_dff(.x,.y)) )



ggplot(df_mean,aes(x=diff_m,y=importance_a))+geom_line()+ theme_bw()+xlab('mean diffe
rence')+ylab('statistic power')
```



Q3

```
standard_dff<-function(s0,s1){


num_trials<-10000
n_0<-30
n_1<-30
mu_0<-3
mu_1<-4
sigma_0<-s0
sigma_1<-s1
alpha<-0.05
set.seed(0)


result<-data.frame(trial=seq(num_trials)) %>%
  mutate(sample_0 = map(.x=trial,.f =~ rnorm(n=n_0,mean=mu_0,sd=sigma_0)), sample_1 =
map(.x=trial,.f =~ rnorm(n=n_1,mean=mu_1,sd=sigma_1))) %>%
  mutate(p_value=pmap(.l = list(trial,sample_0,sample_1), .f =~ t.test(..2, ..3, var.
equal = TRUE)$p.value)) %>% mutate(reject_null = p_value<alpha ) %>%
  pull(reject_null) %>%
  mean()

return(result)


}
```

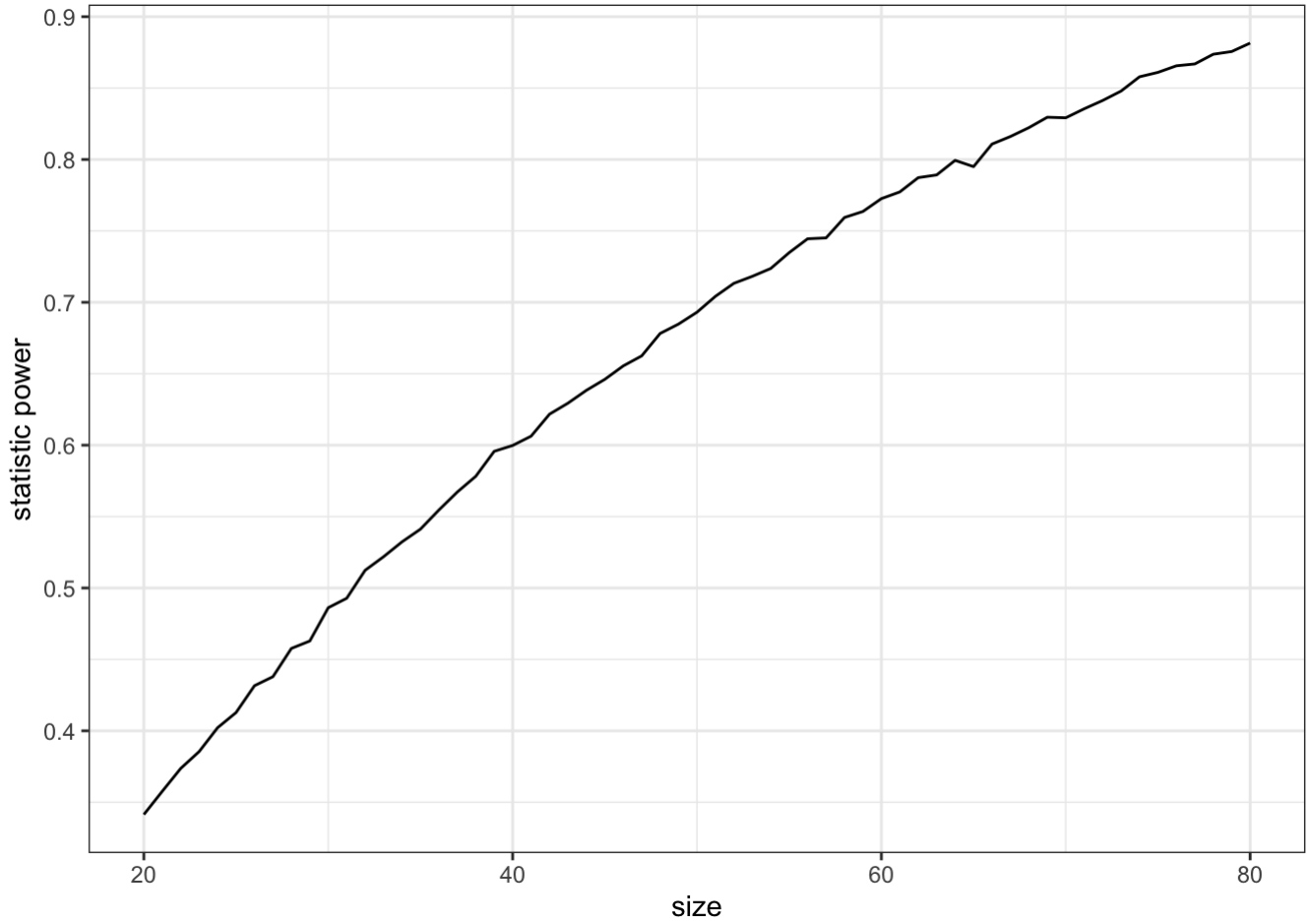
```
s0<-seq(1, 10, 0.5)
diff_m<-seq(1,10,0.5)
s1<-s0+diff_m


df_size<-data.frame(s0,s1,diff_m)
df_stad<-df_size%>%
  mutate(importance_a=map2_dbl(s0,s1, ~standard_dff(.x,.y)) )



ggplot(df_stad,aes(x=s0,y=importance_a))+geom_line()+ theme_bw()+xlab('sd')+ylab('sta
tistic power')
```



Q4

```
size_dff<-function(n){

num_trials<-10000
n_0<-n
n_1<-n
mu_0<-3
mu_1<-4
sigma_0<-2
sigma_1<-2
alpha<-0.05
set.seed(0)


result<-data.frame(trial=seq(num_trials)) %>%
  mutate(sample_0 = map(.x=trial,.f =~ rnorm(n=n_0,mean=mu_0,sd=sigma_0)), sample_1 =
map(.x=trial,.f =~ rnorm(n=n_1,mean=mu_1,sd=sigma_1))) %>%
  mutate(p_value=pmap(.l = list(trial,sample_0,sample_1), .f =~ t.test(..2, ..3, var.
equal = TRUE)$p.value)) %>% mutate(reject_null = p_value<alpha ) %>%
  pull(reject_null) %>%
  mean()

return(result)


}
```

```
n<-seq(20, 80, 1)

df_size<-data.frame(n)
df_se<-df_size%>%
  mutate(importance_a=map_dbl(n, ~size_dff(.x)) )



ggplot(df_se,aes(x=n,y=importance_a))+geom_line()+ theme_bw()+xlab('size')+ylab('stat
istic power')
```


0.9
0.8
0.7
0.6
0.5
0.4
statistic power
20
40
60
80
size