A7: Sample Size, Proportions, Rates, Linear Regression

EPIB 607 - FALL 2021

your name and McGill ID

compiled on November 26, 2021

1 (25 points) REGEN-COV Antibody Combination and Outcomes in Outpatients with Covid-19

(a)

```
## ---- Question-1 ------------------------------------------------------------
set.seed(1240)
power_dist_1a <- replicate(10000, expr = {

  day1 <- rnorm(20, mean = 0, sd = 2.1)
  day22 <- rnorm(20, mean = 1.91, sd = 2.1)

  t.test(day1, day22)$p.value < 0.05

})

tab1a <- prop.table(table(power_dist_1a)); tab1a
```

```
## power_dist_1a
##  FALSE   TRUE
## 0.1981 0.8019
```

Percentage of samples that would result in a p_value less than 0.05 (i.e statistical significance == rejecting the null) is 80.2% which shows that the study is powered at 80% to detect the desired effect.

(b)

```
power_dist_1b <- replicate(10000, expr = {

  day1 <- rnorm(50*0.9, mean = 0, sd = 2.1)
  day22 <- rnorm(50*0.9, mean = 1.25, sd = 2.1)

  t.test(day1, day22)$p.value < 0.05

})

tab1b <- prop.table(table(power_dist_1b)); tab1b
```

```
## power_dist_1b
##  FALSE   TRUE
## 0.1998 0.8002
```

Percentage of samples that would result in a p_value less than 0.05 (i.e statistical significance ==> rejecting the null) is 80% which shows that the study is powered at 80% to detect a difference of 1.25 log10 copies/mL.

(c)

```
library(pwr)

detect_diff_calc <- pwr.t.test(n = 50*0.9, power = 0.8)

# d  Effect size (Cohen's d) -
# difference between the means / the pooled standard deviation
detectable_diff <- detect_diff_calc$d * 3.8
detectable_diff
```

```
## [1] 2.269246
```

At an 80% power with a sample size of 50, dropout rate of 10% and an sd of 3.8 log10 copies/mL the detectable difference is 2.2692457 log10 copies/mL

OR

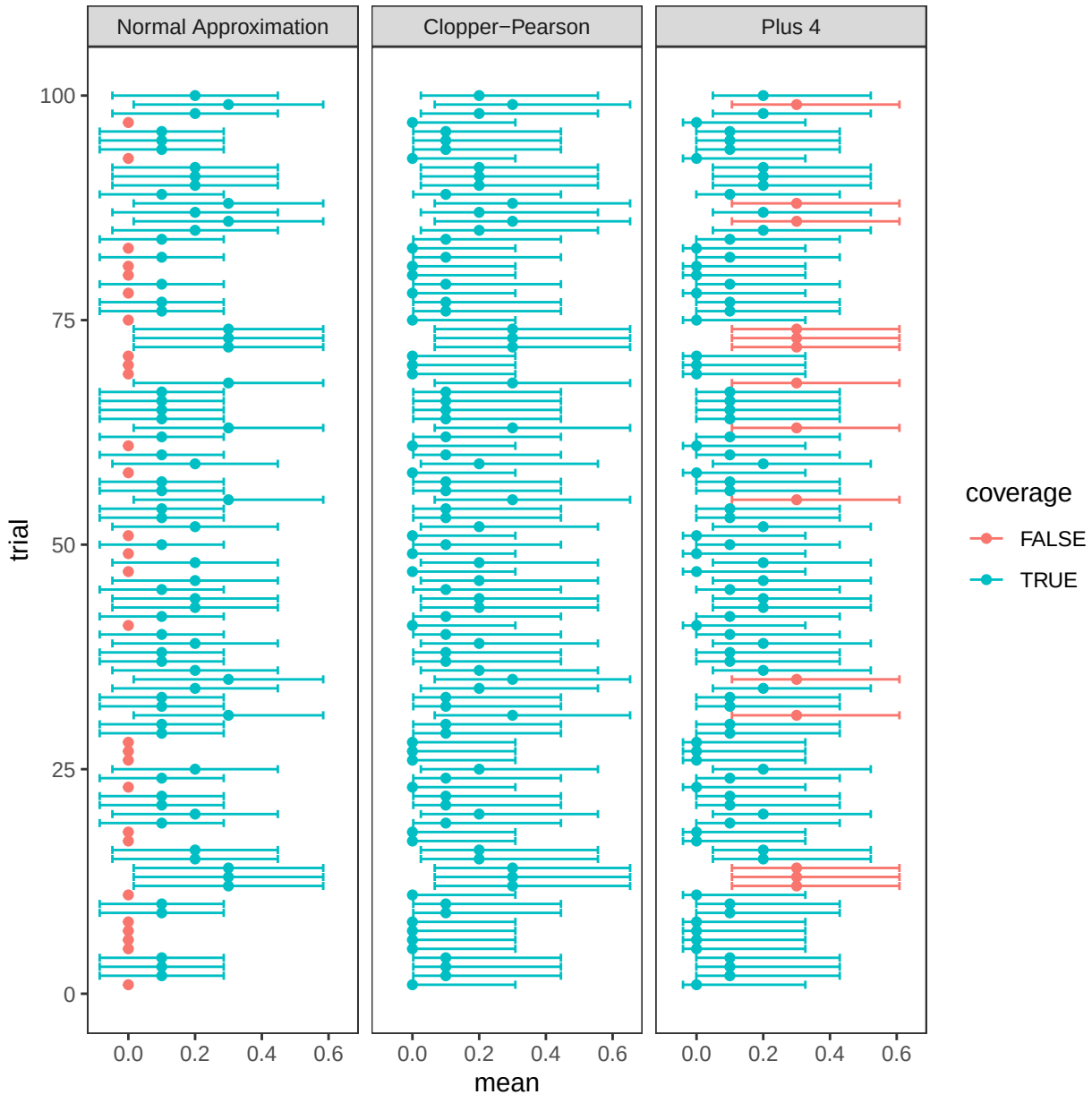
```r
power_calc <- pwr.t.test(n = 50*0.9, d = 2.27/3.8)
power_calc$power
```

```
## [1] 0.8002617
```

At an 80.03% power with a sample size of 50, dropout rate of 10% and an sd of 3.8 log10 copies/mL the detectable difference is 2.27 log10 copies/mL

2 (25 points) Simulation study for confidence intervals of proportions

(a)



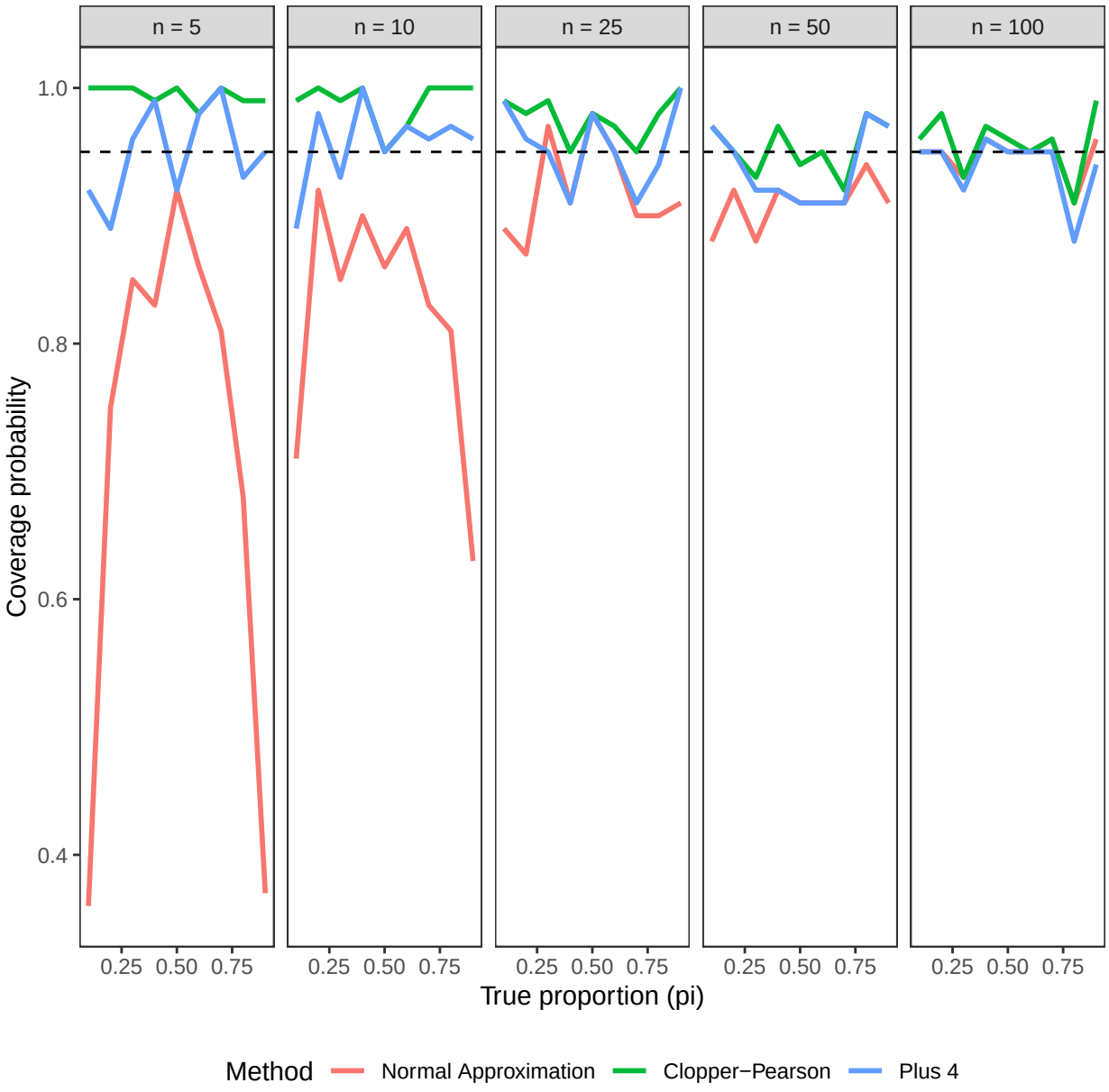
(b)

The coverage probability was 72% for the Normal Approximation, 100% for the exact method (Clopper-Pearson), and 86% for the Plus 4 method. The coverage for the Normal Approximation was much lower than the expected coverage probability of 95%. This was because

any trial with 0 successes had a confidence interval from 0 to 0 using this method, a major limitation of using the normal approximation when the expected number of events is low. The coverage for the exact method was higher than the expected coverage probability of 95%, showing that it is a conservative method for estimating confidence intervals. The coverage for the plus 4 method was between the other two, and closer to the expected probability of 95% than the normal approximation, showing that it is a reasonable and simple way to improve the accuracy of confidence intervals based on the normal approximation for a low expected number of events.

(c)

The coverage probability for the normal approximation increases with increasing number of trials (n) and increasing expected number of events. It does particularly poor with a very low (or high) expected number of events (i.e. when p = 0.1 or 0.9 and n = 5). The coverage probability for the exact method (Clopper-Pearson) decreases with increasing number of trials (n) and increasing expected number of events. The coverage probability for the plus 4 method is fairly similar with increasing number of trials (n) and expected events, at around 95%. While the normal approximation has very low coverage probability at low n and the exact method has very high coverage probability at low n with the plus 4 method in the middle, as n increases the coverage probabilities for each of the three methods converge to around 95%.


n = 5
n = 10
n = 25
n = 50
n = 100
1.0
0.8
0.6
0.4
Coverage probability
0.25 0.50 0.75
True proportion (pi)
Method
Normal Approximation
Clopper−Pearson
Plus 4

3 (25 points) Concordance between PCR-based extraction-free saliva and nasopha- ryngeal swabs for SARS-CoV-2 testing - PART I

(a)

Let $X_{i,N}$ be the Ct value of i^{th} individual from `Nasopharyngeal`(NPS), and $X_{i,S}$ be the Ct value of i^{th} individual from `Saliva`. Denote the population mean of two groups by μ_N and μ_S (1 mark). The question asks us to test $\mathcal{H}_0 : \mu_N = \mu_S$ v.s. $\mathcal{H}_1 : \mu_N \neq \mu_S$ (1 mark). The parameter of interest is the difference: $\mu_N - \mu_S$. (1 mark) We noticed that the observations in `Nasopharyngeal` and `Saliva` are **paired** by `ID`. (1 mark) Conventionally, a paired-t test will be considered to study the group mean difference. The corresponding linear regression model would be:

$$\mathbb{E}(\texttt{Nasopharyngeal}) = \beta_0 + 1 * \texttt{Salive2 marks}$$

By constraining the coefficients of `Salive` to be 1, the intercept β_0 now become the $\mu_N - \mu_S$, and hence we can test the group mean difference by testing $\mathcal{H}_0 : \beta_0 = 0$ v.s. $\mathcal{H}_1 : \beta_0 = neq0$. (2 marks)

Regression model and result will be found in part(b).

(b)

```
##
## Call:
## lm(formula = Saliva ~ offset(1 * Nasopharyngeal), data = dt_symp)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.2663  -4.3815  -0.4367   4.3630  19.4285
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.1329     0.7499   0.177     0.86
##
## Residual standard error: 6.749 on 80 degrees of freedom
##   (46 observations deleted due to missingness)
```

The estimated intercept $\hat{\alpha} \approx 0.13$, with standard error 0.75 (1 mark). **For the same**

participant, the Ct value from Saliva is **on average** 0.13 unit higher than the Ct value from NPS.(2 marks)

The p-value for testing $\mathcal{H}_0 : \beta_0 = 0$ is 0.86, which is not significant at 0.05 level.(1 mark) It shows that we do not have enough evidence to reject the null hypothesis of two groups having equal mean. (1 mark)

(c)

The p-value produced by the constrained linear regression is the same as a paired-t test. Hence the β_0 is estimated by the mean of sample differences. Take the difference $X_{i,d} = X_{i,N} - X_{i,S}$, we have a new sample of X_d. Noted that the difference can only be computed when both measurements are available for the same individual, the sample size for complete cases is $m = 81$:

```
dt_symp_complete = dt_symp[complete.cases(dt_symp), ]
nrow(dt_symp_complete)
```

```
## [1] 81
```

$$\widehat{\beta_0} = \frac{1}{m}\sum_{i=1}^{m} X_{i,d} \approx 0.13$$

```
diff = dt_symp_complete$Saliva - dt_symp_complete$Nasopharyngeal
beta0 = mean(diff)
print(beta0)
```

```
## [1] 0.1329084
```

The standard error is calculated from sampling distribution:

$$SE(\widehat{\beta_0}) = \frac{SD_d}{\sqrt{m}} = \sqrt{\frac{1}{m-1}\sum_{i=1}^{m}(X_{i,d} - \widehat{\beta_0})} \Big/ \sqrt{n} \approx 0.75$$

```
std.dev_d = sd(diff)
se_beta0 = std.dev_d/sqrt(length(diff))
print(se_beta0)
```

```
## [1] 0.7498556
```

where SD_d is the sample standard deviation of $\{X_{1,d}, X_{2,d}, \cdots, X_{m,d}\}$. We then compute the t-score

$$T = (\widehat{\beta_0} - 0)/SE(\widehat{\beta_0}) \approx 0.18$$

and obtain the p-value from t-distribution with degree of freedom $m - 1 = 80$.

```
t.score = (beta0 - 0)/se_beta0
p.val = pt(q = t.score, df = length(diff) - 1,lower.tail = F)*2
print(c(t.score, p.val))
```

```
## [1] 0.1772453 0.8597637
```

One easy way to validate our calculation:

```
t.test(x=dt_symp_complete$Saliva, y=dt_symp_complete$Nasopharyngeal, alternative = "two.
```

```
##
##  Paired t-test
##
## data:  dt_symp_complete$Saliva and dt_symp_complete$Nasopharyngeal
## t = 0.17725, df = 80, p-value = 0.8598
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.359352  1.625169
## sample estimates:
## mean of the differences
##               0.1329084
```

(d)

A permutation test is based on the fact that, by randomly shuffling the observation in NPS while keeping the observation in Saliva(and of course the subject ID) the same, the association between two columns (NPS and Saliva) is broken. Hence, repeating the permutation multiple times will provide us with many samples that is under the null hypothesis of no difference in group mean. And we can use the mean differences from these samples to form an empirical null distribution, and evaluate the observed sample mean difference against it (2 marks).

```
permutation.test <- function(x1, x2, perm){
  distribution=c()
  result=0
  original = mean(x1-x2)
  for(i in 1:perm){
```

```
    distribution[i]=mean(x1 - sample(x2, size = length(x2), replace = FALSE) )
    }
  result=sum(abs(distribution) >= abs(original))/(perm)
  return(list('p-value' = result, 'permutation' = distribution))
}
set.seed(111)
result = permutation.test(x1 = dt_symp_complete$Saliva, x2 = dt_symp_complete$Nasopharyn
print(result$`p-value`)
```

```
## [1] 0.7943
```

The permutation shows a p-value around 0.79(2 marks). It is close to the p-value from part (c), and both are insignificant (1 mark), showing that we do not have enough evidence to reject the null hypothesis that there is no difference in the mean Ct value from Saliva and NPS. The consistency is expected, as we explained the theory of permutation test above. However, by altering the seed, we observed that the p-values ranges from 0.77 to 0.8, always slightly smaller than the p-value from paired t-test (try the following code). There could be some hidden structure in the data, and breaking the structure by permutation results in an under-estimated null distribution tail. We cannot guarantee that the empirical null distribution generated from permutation is the truth, and it can be slightly different from the theoretical null distribution(2 marks).

```
for (i in 1:100){
  set.seed(i)
  result = permutation.test(x1 = dt_symp_complete$Saliva, x2 = dt_symp_complete$Nasophar
  print(result$`p-value`)
  }
```

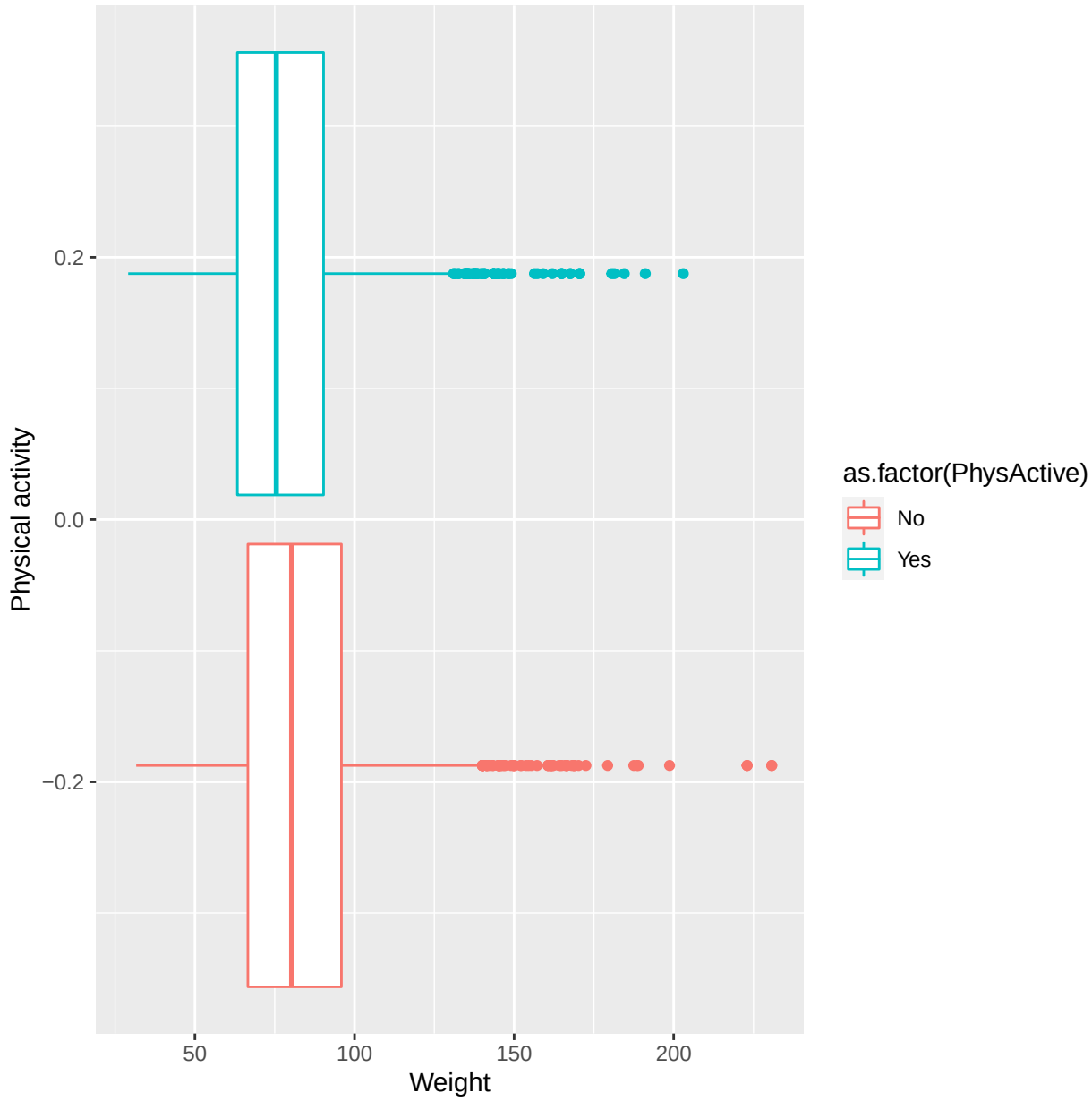
4 (25 points, 5 each) Physical activity in NHANES

```
## # A tibble: 6 x 76
##      ID SurveyYr Gender   Age AgeDecade AgeMonths Race1 Race3 Education
##   <int> <fct>    <fct>  <int> <fct>         <int> <fct> <fct> <fct>
## 1 51624 2009_10  male      34 " 30-39"        409 White <NA>  High School
## 2 51624 2009_10  male      34 " 30-39"        409 White <NA>  High School
## 3 51624 2009_10  male      34 " 30-39"        409 White <NA>  High School
## 4 51625 2009_10  male       4 " 0-9"           49 Other <NA>  <NA>
## 5 51630 2009_10  female    49 " 40-49"        596 White <NA>  Some College
## 6 51638 2009_10  male       9 " 0-9"          115 White <NA>  <NA>
## # ... with 67 more variables: MaritalStatus <fct>, HHIncome <fct>,
## #   HHIncomeMid <int>, Poverty <dbl>, HomeRooms <int>, HomeOwn <fct>,
## #   Work <fct>, Weight <dbl>, Length <dbl>, HeadCirc <dbl>, Height <dbl>,
## #   BMI <dbl>, BMICatUnder20yrs <fct>, BMI_WHO <fct>, Pulse <int>,
## #   BPSysAve <int>, BPDiaAve <int>, BPSys1 <int>, BPDia1 <int>, BPSys2 <int>,
## #   BPDia2 <int>, BPSys3 <int>, BPDia3 <int>, Testosterone <dbl>,
## #   DirectChol <dbl>, TotChol <dbl>, UrineVol1 <int>, UrineFlow1 <dbl>, ...
```

(a)

```
## [1] 4649
```

There are 4649 active samples.



The distribution for both groups are right-skewed with many outliers on the larger side. The median of people without physical activities are slightly larger than that of people with physical activities. The widths of IQR for both groups are similar, and the IQRs largely overlapped.

(b)

objective: investigate whether the difference of weight exists between the physical active and non-active groups.
Model: $\mu = \mu_0 + \beta * I_{active}$
parameter of interest: β, the difference between two groups;

Let set $I_{active} = 0$ if non-active, and $I_{active} = 1$ if active. μ_0 the mean weight of people without physical activities; mu, the expected weight of a person given the physical state.
Null hypothesis (h_0): $\mu = \mu_0$, i.e. $\beta = 0$

(c)

```
##
## Call:
## lm(formula = Weight ~ as.factor(PhysActive), data = active_weight)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -51.336 -15.379  -2.536  12.378 147.764
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)                82.9363     0.3568  232.42   <2e-16 ***
## as.factor(PhysActive)Yes   -4.9575     0.4771  -10.39   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 21.52 on 8254 degrees of freedom
## Multiple R-squared:  0.01291,    Adjusted R-squared:  0.01279
## F-statistic:   108 on 1 and 8254 DF,  p-value: < 2.2e-16
```

The estimated value of μ_0 is 82.94, representing that the estimated mean weight of the population without physical activity is 82.94kg. That of β is -4.96, representing that the physically inactive people are expected to be 4.96kg heavier than people with physical activities.

(d)

```
## [1] -5.892537 -4.022420
```

The 95% confidence interval is (-5.89, -4.02), representing that we are 95% confident that the true value of β will fall in this region. We reject the null hypothesis because the 95% CI does not contain 0, and say that being physically active has a statistically significant negative effect on weight.
To calculate the 95% CI, we assume our samples are simple random samples and the sample size is large enough for the CLT to kick in.

(e)

(Student are free to have covariate in their lm)

```
## The 95%CI is:  -5.902978 -4.070736
```

The 95% CI generated by the bootstrap is very similar to what we get in (d). Therefore, our assumption in (d) is reasonable.

Code

```
## ---- Setup ------------------------------------------------------------------
# set default chunk options here
knitr::opts_chunk$set(
  echo = FALSE,           # don't show code
  warning = FALSE,        # don't show warnings
  message = FALSE,        # don't show messages (less serious warnings)
  cache = TRUE,          # set to TRUE to save results from last compilation
  fig.align = "center",    # center figures
  fig.asp = 1           # fig.aspect ratio
)
library(tidyverse)
library(readxl)
library(here)
library(kableExtra)
## ---- Question-1 -------------------------------------------------------------
set.seed(1240)
power_dist_1a <- replicate(10000, expr = {

  day1 <- rnorm(20, mean = 0, sd = 2.1)
  day22 <- rnorm(20, mean = 1.91, sd = 2.1)

  t.test(day1, day22)$p.value < 0.05

})

tab1a <- prop.table(table(power_dist_1a)); tab1a
power_dist_1b <- replicate(10000, expr = {

  day1 <- rnorm(50*0.9, mean = 0, sd = 2.1)
  day22 <- rnorm(50*0.9, mean = 1.25, sd = 2.1)

  t.test(day1, day22)$p.value < 0.05

})

tab1b <- prop.table(table(power_dist_1b)); tab1b
library(pwr)

detect_diff_calc <- pwr.t.test(n = 50*0.9, power = 0.8)

# d  Effect size (Cohen's d) -
# difference between the means / the pooled standard deviation
```

```r
detectable_diff <- detect_diff_calc$d * 3.8
detectable_diff
power_calc <- pwr.t.test(n = 50*0.9, d = 2.27/3.8)
power_calc$power
## ---- Question-2 ------------------------------------------------------------
## ---- part a -----
set.seed(1234)

# simulate 100 trials from binomial distribution
## get the proportion of successes (number of successes divided by 10)
sim <- replicate(100, {rbinom(n = 10, size = 1, p=0.1) %>%
    sum()/10})

# calculate 95% confidence interval using normal approximation
## lower bound
norm_lower <- sim + qnorm(0.025)*sqrt(sim*(1-sim)/10)
## upper bound
norm_upper <- sim + qnorm(0.975)*sqrt(sim*(1-sim)/10)
## put results in dataframe
q2_ci <- data.frame(trial=seq(1,100), method = "Normal Approximation",
                    mean=sim,
                    lower_95 = norm_lower, upper_95 = norm_upper)

# calculate 95% confidence interval using Clopper-Pearson method
for(i in 1:length(sim)){
  ## lower bound
  cp_lower <- mosaic::binom.test(x=sim[i]*10, n=10,
                                 ci.method=c("Clopper-Pearson"))$conf.int[1]
  ## upper bound
  cp_upper <- mosaic::binom.test(x=sim[i]*10, n=10,
                                 ci.method=c("Clopper-Pearson"))$conf.int[2]
  ## put results in dataframe
  q2_ci <- rbind(q2_ci, data.frame(trial = i, method = "Clopper-Pearson",
                                   mean = sim[i],
                                   lower_95 = cp_lower, upper_95 = cp_upper))
}

# calculate 95% confidence interval using plus 4 method
## add 2 successes and 2 failures to each trial and calculate new proportion
sim_plus4 <- ((sim*10)+2)/(10+4)
## lower bound
plus4_lower <- sim_plus4 + qnorm(0.025)*sqrt(sim_plus4*(1-sim_plus4)/(10+4))
## upper bound
plus4_upper <- sim_plus4 + qnorm(0.975)*sqrt(sim_plus4*(1-sim_plus4)/(10+4))
```

```r
## put results in dataframe
q2_ci <- rbind(q2_ci, data.frame(trial = seq(1,100),
                                 method = "Plus 4",
                                 mean = sim,
                                 lower_95 = plus4_lower,
                                 upper_95 = plus4_upper))

# create indicator variable for coverage in dataframe
q2_ci <- q2_ci %>%
  mutate(coverage = ifelse(lower_95 <= 0.1 & upper_95 >= 0.1, TRUE, FALSE),
         method = factor(method, levels = c("Normal Approximation",
                                            "Clopper-Pearson",
                                            "Plus 4")))

# plot
q2a_plot <- ggplot(data = q2_ci, aes(x=mean, y = trial))+
  geom_errorbarh(aes(xmin=lower_95,xmax=upper_95, col = coverage))+
  geom_point(aes(col = coverage)) +
  facet_grid(cols = vars(method))+
  theme_bw() +
  theme(panel.grid = element_blank())

q2a_plot
## ---- Question-3 ------------------------------------------------------------
## ---- part b -----

# calculate coverage probability for normal approximation
q2_ci %>% filter(method == "Normal Approximation") %>%
  select(coverage) %>% table() %>% prop.table()

# calculate coverage probability for clopper-pearson
q2_ci %>% filter(method == "Clopper-Pearson") %>%
  select(coverage) %>% table() %>% prop.table()

# calculate coverage probability for plus 4
q2_ci %>% filter(method == "Plus 4") %>%
  select(coverage) %>% table() %>% prop.table()
## ---- Question-3 ------------------------------------------------------------
## ---- part c -----

set.seed(4321)

# set combinations of n and pi
df_2c <- data.frame(n = c(5,10,25,50,100, 5, 10, 25, 50),
```

```
                    pi = seq(0.1,0.9, 0.1)) %>%
  tidyr::expand(n, pi)

# create empty dataframe to store results in
sim_2c_df <- data.frame(trial = as.numeric(),
                        method = as.character(),
                        n = as.numeric(),
                        pi = as.numeric(),
                        mean = as.numeric(),
                        lower_95 = as.numeric(),
                        upper_95 = as.numeric())

# run simulations and calculate CIs
simulation_2c <- function(n_arg, pi_arg){
  # simulate 100 trials from binomial distribution
  ## get the proportion of successes (number of successes divided by 10)
  sim <- replicate(100, {rbinom(n = n_arg, size = 1, p=pi_arg) %>%
      sum()/n_arg})

  # calculate 95% confidence interval using normal approximation
  ## lower bound
  norm_lower <- sim + qnorm(0.025)*sqrt(sim*(1-sim)/n_arg)
  ## upper bound
  norm_upper <- sim + qnorm(0.975)*sqrt(sim*(1-sim)/n_arg)
  ## put results in dataframe
  q2_ci <- data.frame(trial=seq(1,100), method = "Normal Approximation",
                      n = n_arg,
                      pi = pi_arg,
                      mean=sim,
                      lower_95 = norm_lower, upper_95 = norm_upper)

  # calculate 95% confidence interval using Clopper-Pearson method
  for(i in 1:length(sim)){
    ## lower bound
    cp_lower <- mosaic::binom.test(x=sim[i]*n_arg, n=n_arg,
                                   ci.method=c("Clopper-Pearson"))$conf.int[1]
    ## upper bound
    cp_upper <- mosaic::binom.test(x=sim[i]*n_arg, n=n_arg,
                                   ci.method=c("Clopper-Pearson"))$conf.int[2]
    ## put results in dataframe
    q2_ci <- rbind(q2_ci, data.frame(trial = i, method = "Clopper-Pearson",
                                     n = n_arg, pi = pi_arg,
                                     mean = sim[i],
                                     lower_95 = cp_lower, upper_95 = cp_upper))
```

```
  }

  # calculate 95% confidence interval using plus 4 method
  ## add 2 successes and 2 failures to each trial and calculate new proportion
  sim_plus4 <- ((sim*n_arg)+2)/(n_arg+4)
  ## lower bound
  plus4_lower <- sim_plus4 + qnorm(0.025)*sqrt(sim_plus4*(1-sim_plus4)/(n_arg+4))
  ## upper bound
  plus4_upper <- sim_plus4 + qnorm(0.975)*sqrt(sim_plus4*(1-sim_plus4)/(n_arg+4))
  ## put results in dataframe
  q2_ci <- rbind(q2_ci, data.frame(trial = seq(1,100),
                                   method = "Plus 4",
                                   n = n_arg,
                                   pi = pi_arg,
                                   mean = sim,
                                   lower_95 = plus4_lower,
                                   upper_95 = plus4_upper))

  # store results in dataframe
  sim_2c_df <<- rbind(sim_2c_df, q2_ci)
}

# run simulation for all combinations of n and pi
mapply(simulation_2c, df_2c$n, df_2c$pi)

# calculate coverage probability
sim_2c_df <- sim_2c_df %>%
  mutate(coverage = ifelse(lower_95 <= pi & upper_95 >= pi, TRUE, FALSE),
         method = factor(method, levels = c("Normal Approximation",
                                            "Clopper-Pearson",
                                            "Plus 4")))

sim_2c_prob <- sim_2c_df %>% group_by(method, n, pi) %>%
  summarize(coverage_prob = sum(coverage == T)/n()) %>% ungroup() %>%
  as.data.frame() %>%
  mutate(n = paste("n =", n),
         n = factor(n, levels = c("n = 5",
                                  "n = 10",
                                  "n = 25",
                                  "n = 50",
                                  "n = 100")))

# create plot
q2c_plot <- ggplot(data = sim_2c_prob, aes(x=pi, y = coverage_prob))+
```

```
  geom_line(aes(col = method), size = 1) +
  geom_hline(yintercept = 0.95, linetype = "dashed")+
  facet_grid(cols = vars(n))+
  xlab("True proportion (pi)")+
  ylab("Coverage probability")+
  scale_color_discrete(name = "Method")+
  theme_bw()+
  theme(panel.grid = element_blank(),
        legend.position = "bottom")
q2c_plot
## ---- Question-3 ----------------------------------------------------------
library(readxl)
library(dplyr)
library(here)

# read symptomatic cohort data
dt_symp <- readxl::read_xlsx(
  here::here("~/Desktop/PhD/EPIB607 TA/doi_10/Ct_values_for_matched_NPS_and_saliva_sampl
  na = "undetected",
  col_names = c("ID","Nasopharyngeal","Saliva"),
  skip = 1,
  col_types = c("text", "numeric","numeric")
) %>%
  dplyr::mutate(cohort = "Symptomatic")

# read asymptomatic cohort data
dt_asymp <- readxl::read_xlsx(
  here::here("~/Desktop/PhD/EPIB607 TA/doi_10/Ct_values_for_matched_NPS_and_saliva_sampl
  na = "undetected",
  col_names = c("ID","Nasopharyngeal","Saliva"),
  skip = 1,
  col_types = c("text", "numeric","numeric")
) %>%
  dplyr::mutate(cohort = "Asymptomatic")
# combine symptomatic and asymptomatic data together
dt <- dplyr::bind_rows(dt_symp, dt_asymp) %>%
  dplyr::mutate(cohort = factor(cohort))
# colMeans(dt[dt$cohort == "Symptomatic", c("Nasopharyngeal", "Saliva")], na.rm = T) #
fit = lm(Saliva ~ offset(1*Nasopharyngeal) , data = dt_symp)
summary(fit)
dt_symp_complete = dt_symp[complete.cases(dt_symp), ]
nrow(dt_symp_complete)
diff = dt_symp_complete$Saliva - dt_symp_complete$Nasopharyngeal
beta0 = mean(diff)
```

```
print(beta0)
std.dev_d = sd(diff)
se_beta0 = std.dev_d/sqrt(length(diff))
print(se_beta0)
t.score = (beta0 - 0)/se_beta0
p.val = pt(q = t.score, df = length(diff) - 1,lower.tail = F)*2
print(c(t.score, p.val))
t.test(x=dt_symp_complete$Saliva, y=dt_symp_complete$Nasopharyngeal, alternative = "two.
permutation.test <- function(x1, x2, perm){
  distribution=c()
  result=0
  original = mean(x1-x2)
  for(i in 1:perm){
    distribution[i]=mean(x1 - sample(x2, size = length(x2), replace = FALSE) )
    }
  result=sum(abs(distribution) >= abs(original))/(perm)
  return(list('p-value' = result, 'permutation' = distribution))
}
set.seed(111)
result = permutation.test(x1 = dt_symp_complete$Saliva, x2 = dt_symp_complete$Nasopharyn
print(result$`p-value`)
for (i in 1:100){
  set.seed(i)
  result = permutation.test(x1 = dt_symp_complete$Saliva, x2 = dt_symp_complete$Nasophar
  print(result$`p-value`)
  }
## ---- Question-4 ----------------------------------------------------------
library(NHANES)
data(NHANES)
head(NHANES)
library(dplyr)
library(ggplot2)

# i. active
sum(na.omit(NHANES$PhysActive == "Yes"))
# ii. plot for physical vs weight
active_weight <- NHANES %>% select(ID, PhysActive, Weight) %>% na.omit()
ggplot(active_weight) + geom_boxplot(aes(Weight, group = PhysActive, color = as.factor(P
regression_summary <- summary(lm(Weight ~ as.factor(PhysActive), data = active_weight))
regression_summary
#Without covariate: 95% CI for PhysActive
regression_summary$coefficients[2,1] + c(-1.96,1.96) * regression_summary$coefficients[2
sample_number = 1:nrow(active_weight)
estimate_coef <- c()
```

```r
for(i in 1:1000){
  temp_list <- sample(sample_number, nrow(active_weight), replace = TRUE)
  temp_regression <- summary(lm(Weight ~ as.factor(PhysActive), data = active_weight[tem
  estimate_coef[i] = temp_regression$coefficients[2,1]
}

cat("The 95%CI is: ", quantile(estimate_coef, c(0.025,0.975)))
```