Problem Set 2:Linear regression – Solutions

Introduction to Statistical Thinking and Data Analysis

MSc in Epidemiology and MSc in Health Data Analytics, Imperial College London

17 October 2022; reviewed 24 October 2022

A) Consolidating concepts

A1. A regression analysis of the outcome blood pressure (measured in units of millimeters of mercury(mmHg)) as a function of weight (measured in kilograms(kg)) resulted in the following regression coefficients: $\beta_0 = 102.25$, $\beta_1 = 0.20$.

This implies that, in our sample:

a) for each 1kg increase in weight, blood pressuer increased by 102.25mmHg.

b) for each 1mmHg increase in blood pressure, weight increased by 0.20kg.

c) for each 0.2kg increase in wight, blood pressure increased by 102.25mmHg.

d) ***for each 1kg in weight, blood pressure increased by 0.20mmHg.***

A2. A residual is defined as:

a) The difference between the actual y values and the mean of y.

b) ***The difference between the actual y values and the predicted y values.***

c) The predicted value of y for the average x value.

d) The square root of the slope.

A3. Regardless of the value of x, the standard deviation of the distribution of y values about the regression line is the same. This assumption of equal standard deviations of residuals about the regression line is called:

a) random error.

b) heteroscedasticity.

c) ***homoscedasticity.***

d) residual analysis.

A4. In the regression equation $y = \beta_0 + \beta_1 x$, β_0 is the:

a) slope of the line.

b) independent variable.

c) ***y intercept.***

d) coefficient of determination.

A5. The equation of a regression line which describes the association of body weight (kg) and plasma volume (litres) of 8 healthy men is: *plasma volume = 0.0857 + 0.0436 * weight.* For a man weighing 85kg, what would you expect their plasma volume to be?

a) ***3.79 litres.***

b) 2.70 litres.

c) 3.71 litres.

d) 3.00 litres.

A6. Least square regression calculates the best fitting line for the observed data by minimising the sum of the squares of the _____ deviations.

a) ***Vertical.***

b) Horizontal.

c) Both of these.

d) None of these.

A7. A simple linear regression is fit to a dataset of 10 samples. How many degrees of freedom are there?

a) 10

b) 9

c) ***8***

d) 7

A8. Which of the following is not an assumption underlying linear regression between outcome variable y and exposure variable x:

a) There is a linear relationship between x and y.

b) The residuals are independent of each other.

c) For any value of x, y is normally distributed.

d) ***The outcome variable is binary.***

A9. The correlation coefficient allows us to measure the strength of association between an outcome and exposure variable. Which of the following is true of the correlation coefficient?

a) A correlation coefficient = 100 indicates perfect positive correlation.

b) The correlation coefficient is independent of the beta1 regression coefficient.

c) ***The correlation coefficient is the number of standard deviations that the outcome variable changes by for a standard deviation change in the exposure variable.***

d) A correlation coefficient = 0 indicates negative correlation.

A10. Which of the following cannot be used to test for departures from normality?

a) Skewness and Kurtosis.

b) Shapiro-Wilk test.

c) Inverse normal plots.

d) ***Logarithmic transformation.***

B) Practicing skills

B1. Revisit the `perulung_ems.csv` dataset from the week 1 tutorial sheet, introduced on page 27 of Kirkwood and Sterne.

Variable	Description
id	Participant ID number
fev1	Forced Expiratory Volume in 1 second
age	Age in years
height	Height in centimeters
sex	Sex (0 = female, 1 = male)
respsymptoms	Presence of respiratory symptoms (0 = no symptoms; 1 = symptoms)

a. Calculate parameter estimates for a linear regression of the outcome FEV1 as a function of height, and the associated standard errors, t-statistics, and p-values using the formulas from Chapter 10 of Kirkwood and Sterne. Plot the fit of your model compared to the data. Check that your calculations agree with the outputs of the `lm(...)` function in R.

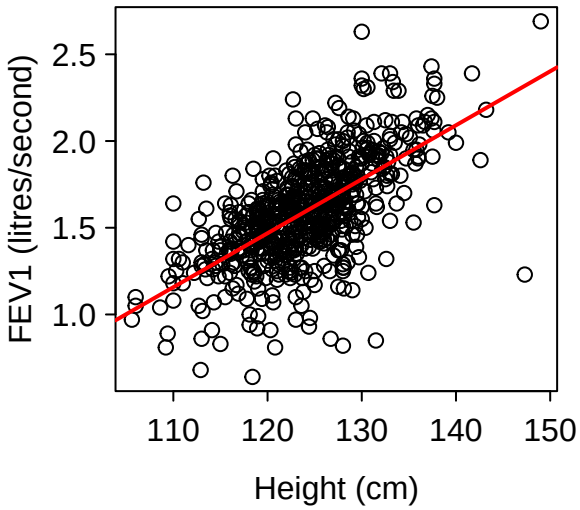
Parameter estimates, standard errors, 95% CI and p-value are in the table (see code below):

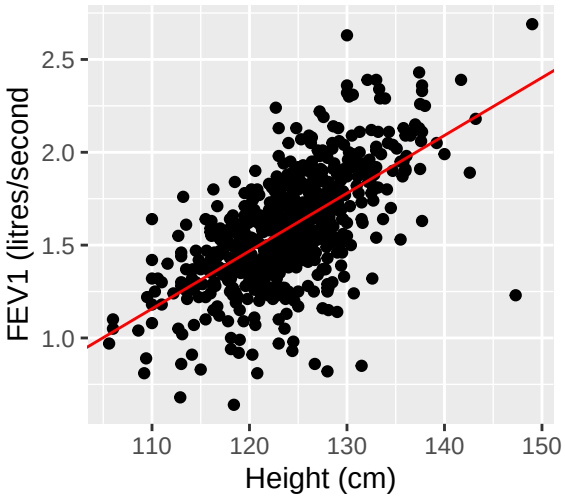
```
data.frame(parameter = c("beta0 (Intercept)", "beta1 (height)",
                         "sigma (residual std. dev.)"),
           estimate = c(beta0_hat, beta1_hat, sigma_hat),
           std_error = c(se_beta0, se_beta1, NA),
           t_stat    = c(beta0_tt, beta1_tt, NA),
           ci_lower  = c(beta0_ci[1], beta1_ci[1], NA),
           ci_upper  = c(beta0_ci[2], beta1_ci[2], NA),
           p_value   = c(beta0_p, beta1_p, NA)) %>%
  knitr::kable(digits = 3)
```

parameter	estimate	std_error	t_stat	ci_lower	ci_upper	p_value
beta0 (Intercept)	-2.266	0.185	-12.216	-2.630	-1.902	0
beta1 (height)	0.031	0.001	20.840	0.028	0.034	0
sigma (residual std. dev.)	0.235	NA	NA	NA	NA	NA

Plot of data and estimated regression line:

```
## Using base R graphics
par(mar = c(3, 3, 0.5, 0.5), tcl = -0.25, mgp = c(2, 0.5, 0))
plot(fev1 ~ height, data = perulung,
     las = 1, xlab = "Height (cm)", ylab = "FEV1 (litres/second)")
abline(a = beta0_hat, b = beta1_hat, col = "red", lwd = 2)

## Using ggplot
ggplot(perulung, aes(height, fev1)) +
  geom_point() +
  geom_abline(slope = beta1_hat, intercept = beta0_hat, color = "red") +
  labs(x = "Height (cm)", y = "FEV1 (litres/second")
```





These estimates exactly match the regression coefficients and residual standard devation estimates produced by the `lm(...)`:

```
    fit1a <- lm(fev1 ~ height, data = perulung)
    summary(fit1a)
```

```
##
## Call:
## lm(formula = fev1 ~ height, data = perulung)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.0881 -0.1329  0.0173  0.1452  0.8503
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.265817   0.185474  -12.22   <2e-16 ***
## height       0.031120   0.001493   20.84   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.2347 on 634 degrees of freedom
## Multiple R-squared:  0.4065, Adjusted R-squared:  0.4056
## F-statistic: 434.3 on 1 and 634 DF,  p-value: < 2.2e-16
```

```
    confint(fit1a)
```

```
##                   2.5 %      97.5 %
## (Intercept) -2.63003408 -1.90160065
## height       0.02818726  0.03405181
```

Code for exercise 1a:

```
## Load data and convert binary outcomes to factors
    library(tidyverse)
    perulung <- read.csv("perulung_ems.csv")
    perulung$sex <- factor(perulung$sex, c(0, 1), c("female", "male"))
    perulung$respsymptoms = factor(perulung$respsymptoms,
                                   c(0, 1), c("no symptoms", "symptoms"))

    ## Calculate regression coefficient estimates
    y <- perulung$fev1
    x <- perulung$height
    ybar <- mean(y)
    xbar <- mean(x)

    beta1_hat <- sum((x - xbar) * (y - ybar)) / sum((x - xbar)^2)
    beta0_hat <- ybar - beta1_hat * xbar

    ## Calculate residual standard deviation
    ypred <- beta0_hat + beta1_hat * x
    rss <- sum((y - ypred)^2)
    df <- length(x) - 2
    sigma_hat <- sqrt(rss / df)

    ## Standard error for regression parameters
    se_beta1 <- sigma_hat / sqrt(sum((x - xbar)^2))
    se_beta0 <- sigma_hat * sqrt(1/length(x) + xbar^2 / sum((x - xbar)^2))

    ## t-statistic, 95% CI, and p-value
    beta0_tt <- beta0_hat / se_beta0
    beta0_ci <- beta0_hat + c(-1,1) * qt(0.975, df) * se_beta0
    beta0_p <- 2 * pt(abs(beta0_tt), df, lower.tail = FALSE)

    beta1_tt <- beta1_hat / se_beta1
    beta1_ci <- beta1_hat + c(-1,1) * qt(0.975, df) * se_beta1
    beta1_p <- 2 * pt(abs(beta1_tt), df, lower.tail = FALSE)

    ## Compare with lm(...)
    fit1a <- lm(fev1 ~ height, data = perulung)
```

b. Interpret the regression coefficients. State the null and alternative hypothesis that is being tested with the p-value reported for the height coefficient and your assessment of this hypothesis.

The estimated best fitting regression equation is:

$$\text{FEV1} = -2.266 + 0.031 \times \text{Height}$$

The estimate for the height coefficient of $\hat{\beta}_1 = 0.031$ (95% CI 0.028, 0.034) indicates that on average FEV1 increases by 0.031 litres/second for each 1 centimeter greater height.

The estimate for the intercept was $\hat{\beta}_0 = -2.266$ (95% CI -2.63, -1.902). This is the predicted value for FEV1 when height is zero centimeters. Interpreting this coefficient is not useful because it is impossible that height is ever zero. In most cases, one should avoid interpreting predictions from a regression model outside of the range of values that are spanned by the data.

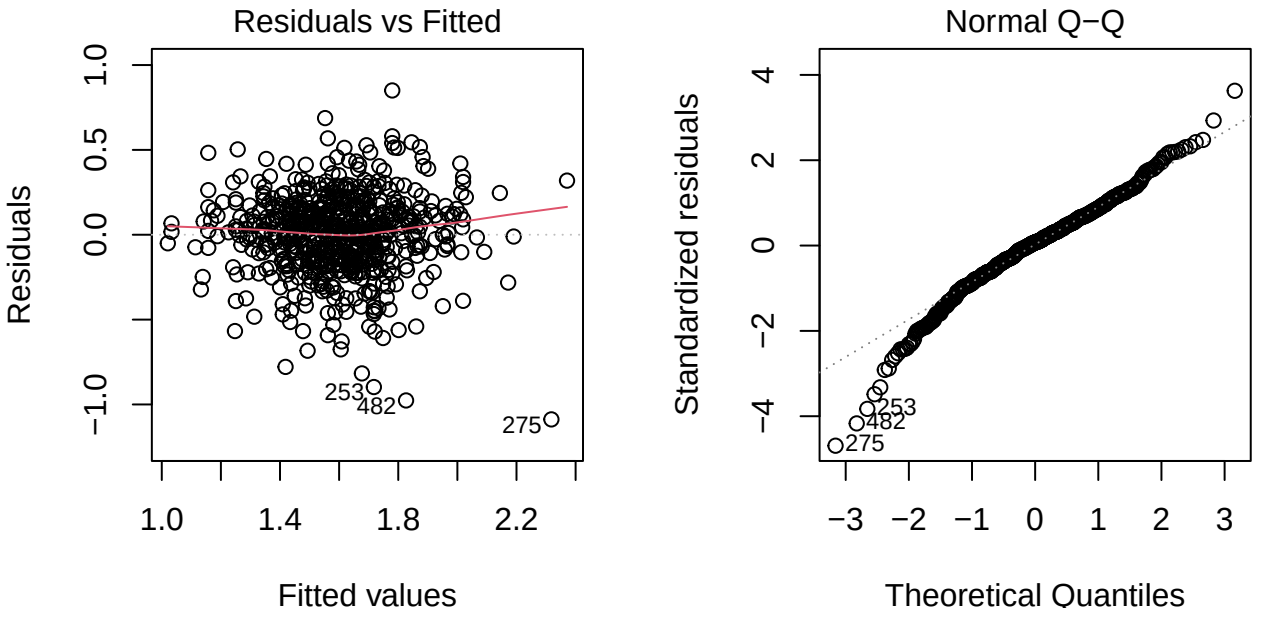
The null hypothesis with which the p-value for the height coefficient is associated is that $\beta_1 = 0$, which implies there is no linear relationship between height and FEV1. The alternative hypothesis is that $\beta_1 \neq 0$—that there is some linear association between FEV1 and height. The value $p < 0.001$ indicates strong evidence against the null hypothesis, and thus we conclude based on our sample that there is a positive association between height and FEV1 amongst children.

c. State and check the assumptions of your regression model—do each of them appear to be satsified?

The four assumptions of our linear regression model are (1) a linear relationship between FEV1 and height, (2) independent observations, (3) normality of residuals, and (4) constant variance of the residuals

Use residual versus fitted value and normal quantile-quantile plots to examine these assumptions:

```
par(mfrow = c(1,2))
plot(fit1a, 1:2)
```



- Linearity: overall this looks pretty good. The smoothed trend fitted to the residuals shows some suggestion of a non-linear pattern towards larger fitted values, but there are few data points in that range.
  - Independence: this is largely based on knowledge of the study design, rather than examination of the residuals. The data consist of a random sample of children in a deprived neighborhood in Lima, Peru. Since children were randomly sampled, we can be confident that the observations are independent.

- Normality: based on the normal Q-Q plot, overall this looks reasonably good, though there is some evidence of a slightly heavy lower tail indicating more extreme negative residuals than we would expect based on the normal distribution.
- Constant variance: Based on the residual versus fitted value plot, this looks fine. There is an ovular cloud of points, with no evidence of an increasing or decreasing spread over the range of fitted values.

d. Convert height into a categorical three groups: height below 120cm, between 120 to 130cm, and above 130cm. Refit your regression model for FEV1 this time using the height categories as your predictor. Interpret the regression coefficient estimates. *Hint: The function* ***cut()*** *will convert a numerical variable to a categorical variable.*

```
perulung$height_cat <- cut(perulung$height, c(-Inf, 120, 130, Inf),
                           labels = c("<=120cm", "120-130cm", ">130cm"))
```

This could also be done using ‘ifelse()‘ statements.

Review the frequency table for the categorical height variable.

```
table(perulung$height_cat)
```

```
##
##   <=120cm 120-130cm    >130cm
##       166       381        89
```

Fit the linear regression model using height as a categorical variable:

```
fit1d <- lm(fev1 ~ height_cat, data = perulung)
summary(fit1d)
```

```
##
## Call:
## lm(formula = fev1 ~ height_cat, data = perulung)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.10135 -0.14135  0.01079  0.16000  1.02079
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          1.37000    0.01928   71.04   <2e-16 ***
## height_cat120-130cm  0.23921    0.02311   10.35   <2e-16 ***
## height_cat>130cm     0.58135    0.03264   17.81   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.2485 on 633 degrees of freedom
## Multiple R-squared:  0.3361, Adjusted R-squared:  0.334
## F-statistic: 160.2 on 2 and 633 DF,  p-value: < 2.2e-16
```

The summary output shows regression coefficients for the height categories `120-130cm` and `>130cm`. No coefficient is reported for the `<=120cm` category because this has been taken as the 'reference' category. The other coefficients report the estimated difference in FEV1 relative to the reference category. Thus we interpret that the average FEV1 for a child 120-130cm is 0.24 litres/second greater than a child below 120cm tall (95% CI 0.19–0.28). The average FEV1 for a child above 130cm is 0.52 (95% CI 0.52–0.66) litres/second greater than for a child below 120cm.

In the case of no other covariates, the `(Intercept)` term is the estimate for the mean FEV1 for the reference category. To see why this is the case, review the estimated regression equation:

$$\text{FEV1} = 1.370 + 0.239 \times \texttt{[height\_cat = 120-130]} + 0.581 \times \texttt{[height\_cat = >130]}.$$

If the height category is <=120cm, then the predicted value for FEV1 will be:

$$\text{FEV1} = 1.370 + 0.239 \times 0 + 0.581 \times 0 = 1.370.$$

If the height category is 120-130cm, then the predicted value for FEV1 will be:

$$\text{FEV1} = 1.370 + 0.239 \times 1 + 0.581 \times 0 = 1.609.$$

If the height category is >130cm, then the predicted value for FEV1 will be:

$$\text{FEV1} = 1.370 + 0.239 \times 0 + 0.581 \times 1 = 1.951.$$

Observe that these categorical linear regression estimates are equal to the sample mean for FEV1 within each height category:

```
    aggregate(fev1 ~ height_cat, perulung, mean)
```

```
##   height_cat     fev1
## 1    <=120cm 1.370000
## 2  120-130cm 1.609213
## 3     >130cm 1.951348
```

e. When analysing categorical predictors, it is often conventional to use the group with greatest frequency as the reference category (though there are other reasonable choices depending on the analysis). Change the reference category for your categorical height variable so that it is the group with the largest number of observations and refit your model. How do the parameter estimates change? What is the interpretation of the new parameters? *Hint: The functions* `levels()` *and* `relevel()` *will be helpful.*

The current reference category can be determined by examining the levels of the factor.

```
    levels(perulung$height_cat)
```

```
## [1] "<=120cm"   "120-130cm" ">130cm"
```

The first level is taken by R as the reference category and dropped from the regression coefficients. The category 120-130cm had the largest number of observations with 381. Use the `relevel()` function to set this as the reference category, and check the change by calling `levels()`.

```
    perulung$height_cat <- relevel(perulung$height_cat, "120-130cm")
    levels(perulung$height_cat)
```

```
## [1] "120-130cm" "<=120cm"   ">130cm"
```

Refit the regression model.

```
    fit1e <- lm(fev1 ~ height_cat, perulung)
    summary(fit1e)
```

```
##
## Call:
## lm(formula = fev1 ~ height_cat, data = perulung)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.10135 -0.14135  0.01079  0.16000  1.02079
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        1.60921    0.01273  126.42   <2e-16 ***
## height_cat<=120cm -0.23921    0.02311  -10.35   <2e-16 ***
## height_cat>130cm   0.34214    0.02925   11.70   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2485 on 633 degrees of freedom
## Multiple R-squared:  0.3361, Adjusted R-squared:  0.334
## F-statistic: 160.2 on 2 and 633 DF,  p-value: < 2.2e-16
```

The regression model output now reports coefficient estimates for the categories `<=120cm` and `>130cm`, with no coefficient reported for the group `120-130cm`.
The term `(Intercept)` is now interpreted as the mean FEV1 for a child with height 120-130cm. The estimate for this parameter is is 1.609, the same as the predicted value for the 120-130cm group in part (d).
The coefficient for the <=120cm category indicates that the expected FEV1 for children with height <=120cm is 0.24 litres/second lower than children with height 120-130cm. Note the the estimate -0.239 is the inverse of the coefficient 0.239 estimated for the difference between the 120-130cm versus <=120cm group in part(d).
The coefficient for the >130cm cateogory indicates that the expected FEV1 for children with height >130cm is 0.34 litres/second greater than the FEV1 for children 120-130cm. Note that the estimate 0.342 is equal to the difference of the coefficients for the >130 and 120-130cm groups in part (d): 0.581 - 0.239 = 0.342.
The estimates for the Residual standard error, degrees of freedom and R^2 are identical to the model fitted in part (d) with a different reference category.

f. Amongst the models estimated in parts (a), (d), and (e), which do you prefer and why?

Based on the exploratory analysis of the sample data and knowledge of the process, it seems plausible that there is a continuous and relatively linear relationship between these variables. Thus the linear trend model fitted in part (a) would be preferred because it describes the relationship between the variables and is more powerful to detect an association.

g. Fit a linear regression model to test the null hypothesis that FEV1 is not associated with respiratory symptoms. What is your conclusions about this hypothesis? How do your effect estimates, statistical inference, and conclusions compare to what you estimated using the equal variance t-test last week?

The estimates from the linear regression and two sample t-test are identical. The coefficient estimate for the term sex = male is the same as the difference of the sample means, and the t-statistic, p-value, and 95% CI for the sex = male coefficient are identical to those estimated by the two-sample t-test with equal variances. Note that the equal variance assumption for the standard t-test is analogous to the constant variance assumption for linear regression.

```
    fit1g <- lm(fev1 ~ respsymptoms, perulung)
    summary(fit1g)
```

```
##
## Call:
## lm(formula = fev1 ~ respsymptoms, data = perulung)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.83917 -0.18917 -0.01876  0.19124  1.06124
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)            1.62876    0.01345 121.054  < 2e-16 ***
## respsymptomssymptoms -0.14959    0.02818  -5.308 1.53e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2981 on 634 degrees of freedom
## Multiple R-squared:  0.04256,    Adjusted R-squared:  0.04105
## F-statistic: 28.18 on 1 and 634 DF,  p-value: 1.531e-07
```

```
    confint(fit1g)
```

```
##                          2.5 %      97.5 %
## (Intercept)           1.602336  1.65517890
## respsymptomssymptoms -0.204920 -0.09425045
```

```
    t.test(perulung$fev1[perulung$respsymptoms == "symptoms"],
           perulung$fev1[perulung$respsymptoms == "no symptoms"],
           var.equal = TRUE)
```

```
##
##  Two Sample t-test
##
## data:  perulung$fev1[perulung$respsymptoms == "symptoms"] and perulung$fev1[perulung$respsymptoms ==
## t = -5.3085, df = 634, p-value = 1.531e-07
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.20492000 -0.09425045
## sample estimates:
## mean of x mean of y
##  1.479172  1.628758
```

B2. In this exercise we will use the NHANES dataset to study child growth by estimating the relationship between height in centimeters and age in months for children aged zero to 10 years (less than 120 months).

Age in months is only reported for children of all ages in the 2009 to 2010 data. For 2011 and 2012, age in months is available only for children aged 0 to 2 years. This is fine for the purposes of our analysis; we will retain the subset of data for which age in months is recorded and below 120 months:

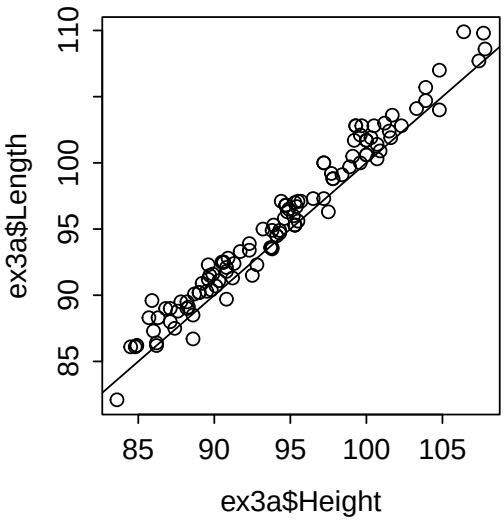
```
library(NHANES)
data(NHANES)
nhanes_child <- subset(NHANES, AgeMonths < 120)
```

Height is measured in two different ways depending on the child age. For children aged 2 and older, standing height is measured and recorded in the variable `Height`. For children aged 0 to 3 years, recumbent length is measured and recorded in the variable `Length`.

a. For children aged 24 to 47 months, both standing height (cm) and recumbent length (cm) were measured. Are standing height and recumbent length equivalent measures of height? Articulate, conduct, and report a hypothesis test to assess this question and report the estimated magnitude of any difference.

Standing height and recumbent length were measured for 116 children aged 24 to 47 months. The mean height was 94.3cm (standard devation 5.8cm) and the average recumbent length was 95.5cm (sd 6.0cm) and the two measures were highly correlated (correlation = 0.98). We used a paired t-test to evaluate the null hypothesis of no difference between height measured as standing height or recumbent length. Recumbent length was on average 1.15cm (95% CI 0.96–1.35cm) greater than standing height. The estimated t-value was 11.8 on 115 degrees of freedom for a two-sided p-value <0.001. This indicates that recumbent length results in on average greater measure of height than standing height for children aged 24 to 47 months.

```
ex3a <- subset(nhanes_child, !is.na(Height) & !is.na(Length))
nrow(ex3a)
summary(ex3a[c("AgeMonths", "Height", "Length")])
sd(ex3a$Height)
sd(ex3a$Length)
cor(ex3a$Height, ex3a$Length)

par(mar = c(3, 3, 0.5, 0.5), tcl = -0.25, mgp = c(2, 0.5, 0))
plot(ex3a$Height, ex3a$Length)
abline(a = 0, b = 1)

t.test(ex3a$Length, ex3a$Height, paired = TRUE)
```



b. Proceed with your analysis assuming recumbent length is a reasonably good approximation for height for children under age 2. Create a single height variable that takes the variable `Length` for children under age 24 months and the variable `Height` for children age 24 to 119 months. Fit a linear regression model to estimate the relationship between age in months and height and give interpretations of both parameters. *Hint: Try the function* ***`ifelse()`****.*

```
nhanes_child$height_all <- ifelse(nhanes_child$AgeMonths < 24,
                                  nhanes_child$Length,
                                  nhanes_child$Height)
fit2b <- lm(height_all ~ AgeMonths, nhanes_child)
summary(fit2b)
```

```
##
## Call:
## lm(formula = height_all ~ AgeMonths, data = nhanes_child)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -22.1899  -3.7007   0.4199   3.9164  16.7669
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 68.500365   0.383144   178.8   <2e-16 ***
## AgeMonths    0.647735   0.006122   105.8   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.238 on 789 degrees of freedom
##   (9 observations deleted due to missingness)
## Multiple R-squared:  0.9342, Adjusted R-squared:  0.9341
## F-statistic: 1.119e+04 on 1 and 789 DF,  p-value: < 2.2e-16
```

The estimated regression equation is:

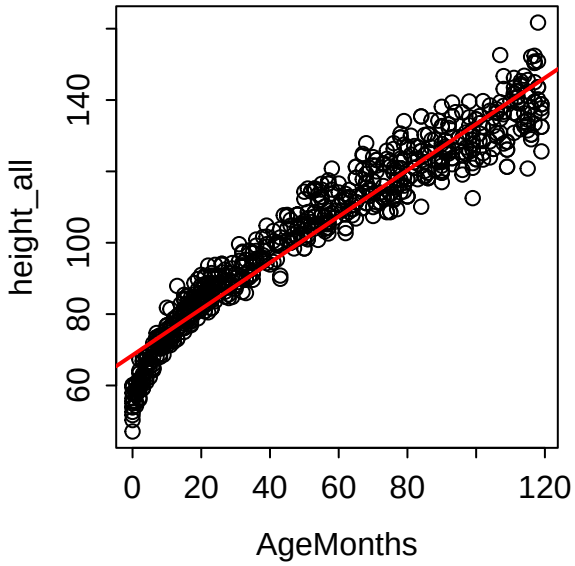
$$\text{Height (cm)} = 68.50 + 0.648 \times \text{Age (months)}.$$

The intercept term 68.5 is the estimated height in centimeters at birth, when age is equal to zero. (Note that while the intercept term is not typically meaningfully interpreted, in this case it does make sense to interpret because age at birth is a meaningful outcome and our dataset includes children down to age zero months.)

The slope 0.648 indicates an average height increase of 0.648cm per month of age.

The figure below shows the estimated regression line compared to the data.

```
par(mar = c(3, 3, 0.5, 0.5), tcl = -0.25, mgp = c(2, 0.5, 0))
plot(height_all ~ AgeMonths, nhanes_child)
abline(fit2b, col = "red", lwd = 2)
```

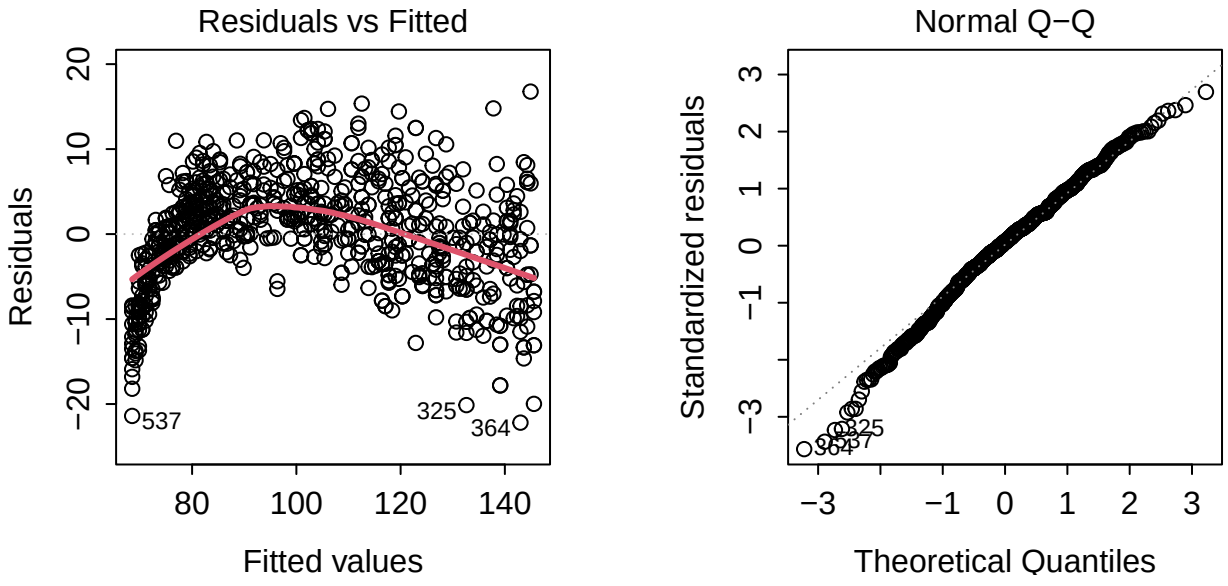


c. Check the assumptions of your regression model fitted in part (b). Do any of the assumptions appear to be violated?

The plot of the estimated regression line to the observed data above gives some indication that the regression line does not optimally describe the data.

The residuals vs. fitted value plot and normal Q-Q

```
par(mfrow = c(1,2), tcl = -0.25, mgp = c(2, 0.5, 0))
plot(fit2b, 1, lwd = 3); plot(fit2b, 2)
```



- **Linearity:** The residuals vs. fitted value plot shows strong evidence of a non-linear trend to the residuals.
- **Independence:** Study participants were randomly sampled from the target population, which should satisfy the independence assumption.
- **Normality:** The curvature of the normal Q-Q plot suggests a left (negatively) skewed distribution, deviating from the normality assumption.
- **Constant variance:** The spread of the residuals increases with larger fitted values.

In week 3 we will look at how transforming variables and robust standard errors can help overcome violated regression assumptions.

C) Advanced learning

Note: These questions are challenging and requires some R tools above a beginner's level and above the level expected to be successful in ISTDA. Make an effort to complete these questions if you would like to challenge your understanding of the central limit theorem, confidence intervals and regression assumptions but do not be discouraged if you are not able to complete them.

C1. *Distribution of the sample mean and confidence intervals.* This exercise will use the NHANES dataset as 'true' population from which to simulate samples and study the properties of the sample mean as an estimator of the population mean. As with last week, we will only consider the subset of the sample who are adults aged 20 years and older. First, lets revisit the the three variables used in question B3 from last weeks problem set:

- Height: Standing height in centimeters.
- BMI: Body Mass Index
- AlcoholYear: Number of days over the past year that participant drank alcoholic beverages

Next, create a vector of responses removing any `NA` values.

```
nhanes20pl <- NHANES[NHANES$Age >= 20, ]
height <- nhanes20pl$Height[!is.na(nhanes20pl$Height)]
   bmi <- nhanes20pl$BMI[!is.na(nhanes20pl$BMI)]
   alcohol <- nhanes20pl$AlcoholYear[!is.na(nhanes20pl$AlcoholYear)]
```

Do the following steps for each variable:

a. Simulate a large number of samples with replacement of size 5, 10, 25, 50, 100, and 500. See the function `?sample`. (The function `?replicate` might also be helpful for simulating a large number of samples.)

b. For each sample, calculate the sample mean, the large-sample 95% CI using the normal distribution, and the small-sample 95% CI using the t-distribution. (The function `?apply` might be useful.)

c. Plot the distribution of the sample mean for each of the increasing sample size. What do you notice about the distribution of the sample mean with increasing sample size for each variable?

The distribution of the sample mean becomes more concentrated around the true population value as the sample size increases.

For height, which follows a normal distribution, the sample mean is also normally distributed for all sample sizes.

For BMI, which has a slight positive skew, the sample mean also has a slight positive skew but becomes normally distributed for n = 25.

For alcohol days per year, which is severely positively skewed, the sample mean is also very skewed at small sample sizes n=5 and n=10.

d. Calculate the coverage of the 95% confidence intervals by calculating for what proportion of samples the 95% confidence interval contains the true population means (calculated in exercise B2 (b) in problem set 1, last week:

	mean	sd
height	168.8	10.1
bmi	28.8	6.7
alcohol_year	75.7	103.6

Create a figure to visually summarise the analysis of 95% CI coverage at different sample sizes, for the large-sample and small-sample CI, and for different variables.

i. How do the coverage of the large-sample CI (using normal distribution) and small-sample CI (using t-distribution) compare at different sample sizes?
ii. For a given sample size, how does the coverage compare for each of the three variables? Can you relate this to your findings in exercise B3 (d) from week 1?

For height, both the normal and t-distribution based confidence intervals attain the nominal 95% coverage when n>50. The calibration of the t-distribution confidence intervals is perfect (equal to 95%) for all sample sizes.

For BMI, which is only slightly non-normally distributed, both attain 95% coverage for n>100 (and close for n>50). At sample sizes $n \leq 10$, the small sample confidence interval is below the nominal 95% level, but very close at 93.6%.

For alcohol consumption days per year, the CI coverage attains 95% for $n = 500$ and is above 93% for $n \geq 100$. However, at small samples $n \leq 10$, the coverage of the t-distribution based CI is better than the coverage of the normal distribution based CI, but is substantially below the nominal 95% level. This is because the t-distribution approximation relies on assumption that the outcome is normally distributed in the population which is severely violated for the number of alcohol days per year outcome.

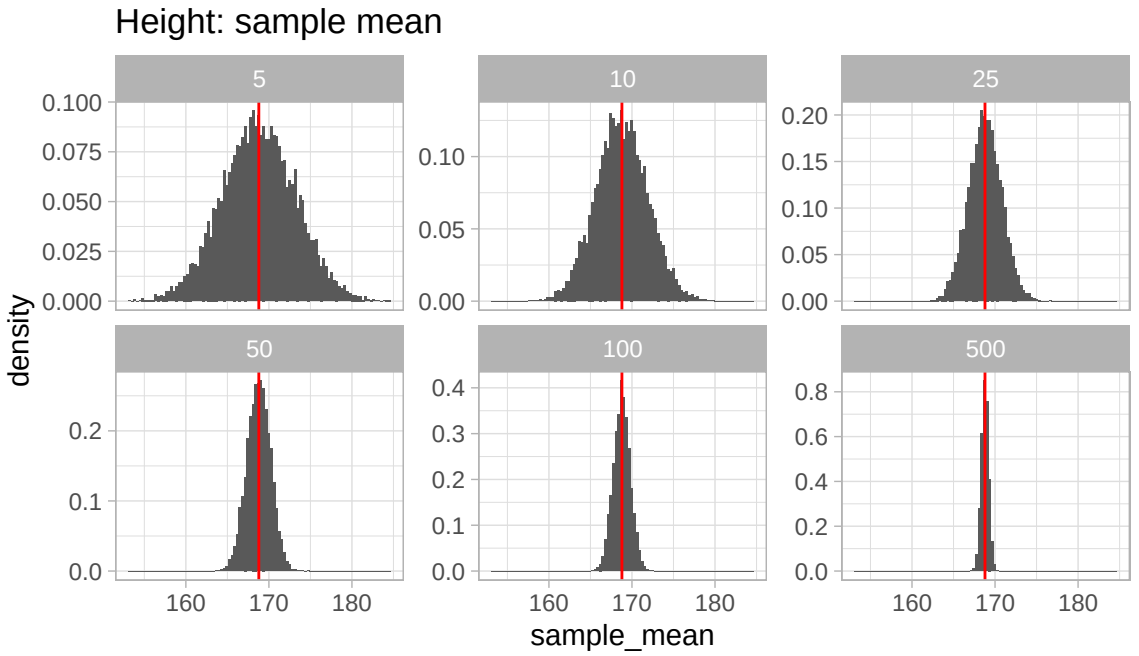
Code to answer question C1:

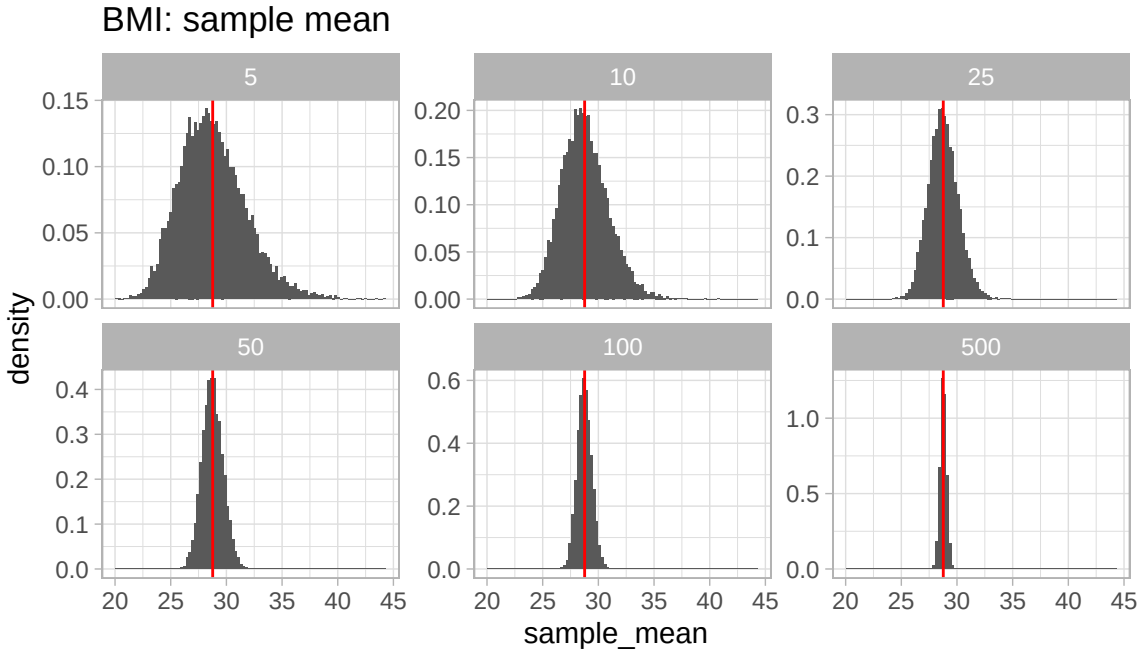
```
set.seed(77316870)

nsim <- 10000 # number of simulated datasets
nsamp <- c(5, 10, 25, 50, 100, 500)   # sample size

sim_ex3 <- function(x, n) {

  mu <- mean(x)  # true population mean
  samp <- replicate(nsim, sample(x, n, replace = TRUE))

  est <- colMeans(samp)
  se <- apply(samp, 2, sd) / sqrt(n)

  ci_l_z <- est - qnorm(0.975) * se
  ci_u_z <- est + qnorm(0.975) * se

  ci_l_t <- est - qt(0.975, n-1) * se
  ci_u_t <- est + qt(0.975, n-1) * se

  list(est = est,
       cover_z = mean(ci_l_z < mu & mu < ci_u_z),
       cover_t = mean(ci_l_t < mu & mu < ci_u_t))
}

height_res <- list()
bmi_res <- list()
alc_res <- list()

for(n in nsamp){
  height_res[[as.character(n)]] <- sim_ex3(height, n)
  bmi_res[[as.character(n)]] <- sim_ex3(bmi, n)
```

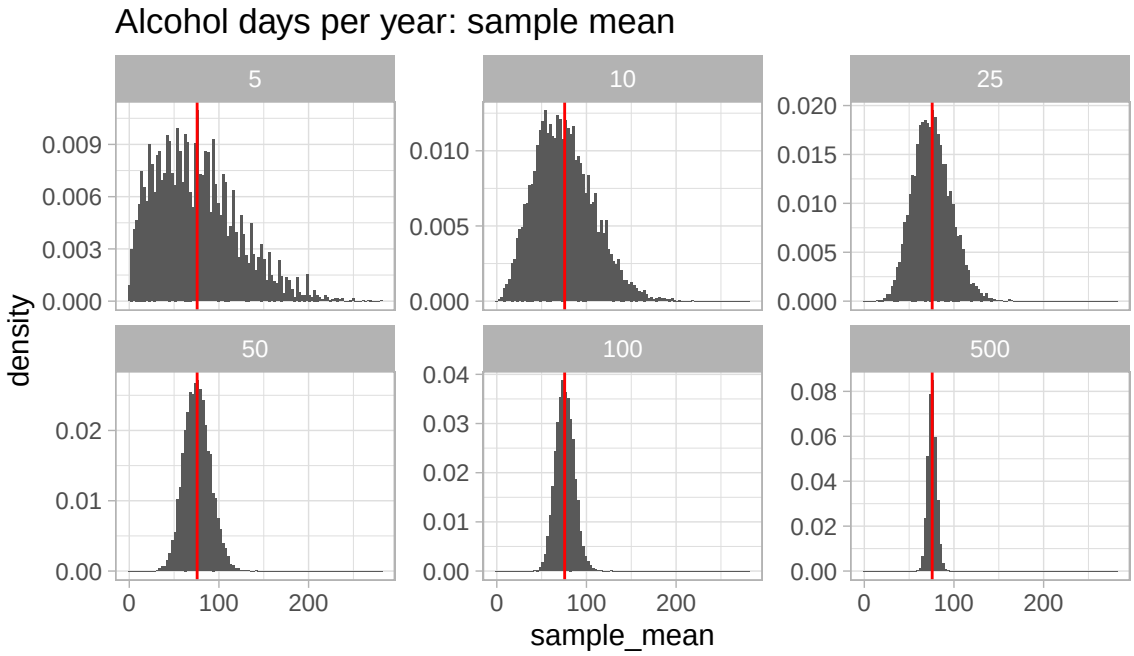
```
  alc_res[[as.character(n)]] <- sim_ex3(alcohol, n)
}
```

Part (c):

```
## sample mean summary
sample_mean_summary <-
  bind_rows(
    data.frame(outcome = "height",
               sapply(height_res, "[[", "est"),
               check.names = FALSE, stringsAsFactors = FALSE),
    data.frame(outcome = "bmi",
               sapply(bmi_res, "[[", "est"),
               check.names = FALSE, stringsAsFactors = FALSE),
    data.frame(outcome = "alcohol",
               sapply(alc_res, "[[", "est"),
               check.names = FALSE, stringsAsFactors = FALSE)
  ) %>%
  gather(n, sample_mean, `5`:`500`) %>%
  mutate(outcome = fct_relevel(outcome, "height", "bmi", "alcohol"),
         n = as.integer(n))

sample_mean_summary %>%
  filter(outcome == "height") %>%
  ggplot(aes(sample_mean)) +
  geom_histogram(aes(y = ..density..), bins = 100) +
  geom_vline(xintercept = mean(height), color = "red") +
  facet_wrap(~n, scales = "free_y") +
  theme_light() +
  ggtitle("Height: sample mean")

sample_mean_summary %>%
  filter(outcome == "bmi") %>%
  ggplot(aes(sample_mean)) +
  geom_histogram(aes(y = ..density..), bins = 100) +
  geom_vline(xintercept = mean(bmi), color = "red") +
  facet_wrap(~n, scales = "free_y") +
  theme_light() +
  ggtitle("BMI: sample mean")

sample_mean_summary %>%
  filter(outcome == "alcohol") %>%
  ggplot(aes(sample_mean)) +
  geom_histogram(aes(y = ..density..), bins = 100) +
  geom_vline(xintercept = mean(alcohol), color = "red") +
  facet_wrap(~n, scales = "free_y") +
  theme_light() +
  ggtitle("Alcohol days per year: sample mean")
```


Height: sample mean
5
10
25
50
100
500
density
sample_mean


BMI: sample mean
5
10
25
50
100
500
density
sample_mean



Part (d):

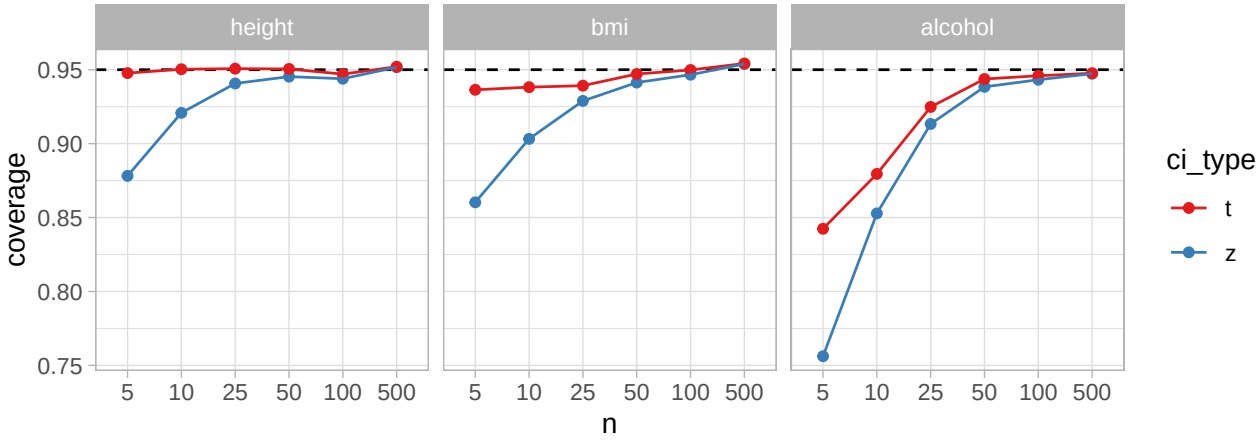
```
## Compile coverage estimates
cover_summary <-
  bind_rows(
    tibble(outcome = "height",
           n = nsamp,
           ci_type = "z",
           coverage = sapply(height_res, "[[", "cover_z")),
    tibble(outcome = "bmi",
           n = nsamp,
           ci_type = "z",
           coverage = sapply(bmi_res, "[[", "cover_z")),
    tibble(outcome = "alcohol",
           n = nsamp,
           ci_type = "z",
           coverage = sapply(alc_res, "[[", "cover_z")),
    tibble(outcome = "height",
           n = nsamp,
           ci_type = "t",
           coverage = sapply(height_res, "[[", "cover_t")),
    tibble(outcome = "bmi",
           n = nsamp,
           ci_type = "t",
           coverage = sapply(bmi_res, "[[", "cover_t")),
    tibble(outcome = "alcohol",
           n = nsamp,
           ci_type = "t",
           coverage = sapply(alc_res, "[[", "cover_t")),

  ) %>%
  mutate(outcome = fct_relevel(outcome, "height", "bmi", "alcohol"))
```

```r
## summary table
cover_summary %>%
  unite(key, outcome, ci_type) %>%
  spread(key, coverage) %>%
  select(n, height_z, height_t, bmi_z, bmi_t, alcohol_z, alcohol_t) %>%
  knitr::kable(caption = "Coverage of theoretical 95% confidence intervals")
```

Table 4: Coverage of theoretical 95% confidence intervals

n	height_z	height_t	bmi_z	bmi_t	alcohol_z	alcohol_t
5	0.8782	0.9477	0.8603	0.9364	0.7562	0.8424
10	0.9208	0.9503	0.9032	0.9382	0.8528	0.8795
25	0.9407	0.9507	0.9289	0.9392	0.9133	0.9248
50	0.9453	0.9505	0.9413	0.9471	0.9384	0.9437
100	0.9439	0.9470	0.9466	0.9498	0.9432	0.9459
500	0.9517	0.9520	0.9539	0.9542	0.9473	0.9476

```r
## summary figure
cover_summary %>%
  mutate(n = as.factor(n)) %>%
  ggplot(aes(n, coverage, color = ci_type, group = ci_type)) +
  geom_hline(yintercept = 0.95, linetype = "dashed") +
  geom_point() +
  geom_line() +
  scale_color_brewer(palette = "Set1") +
  facet_wrap(~outcome) +
  theme_light()
```



C2). *Consequences of violating regression assumptions.* In exercise B2, we demonstrated that regressing child height on age in months did not satisfy several of the linear regression assumptions. In this exercise, we will use simulation to explore the consequences of this for our statistical inference.

Similarly to exercise C1 above, this exercise will use the `nhanes_child` dataset as a 'true' population from which to simulate smaller samples and study the properties of statistical inference for linear regression coefficient estimates. Consider the regression parameter estimates using the full dataset in exercise B2 b. as true population values for β_0 and β_1.

Do the following steps:

a. Simulate smaller datasets by sampling rows from the `nhanes_child` dataset. Sample a large number of datasets with replacement of size 10, 25, 50, 100, and 500. Since both the outcome height and covariate AgeMonths are needed, entire rows must be resampled rather than simply resampling values from a vector. Instead of using the function `sample()`, use `sample.int()` to randomly sample rows to retain, and then subset the data frame to only these rows. Example code for generating one simulated dataset:

```
maxrow <- nrow(nhanes_child)
n <- <sample sizze>
df <- nhanes_child[sample.int(maxrow, n, replace = TRUE), ]
```

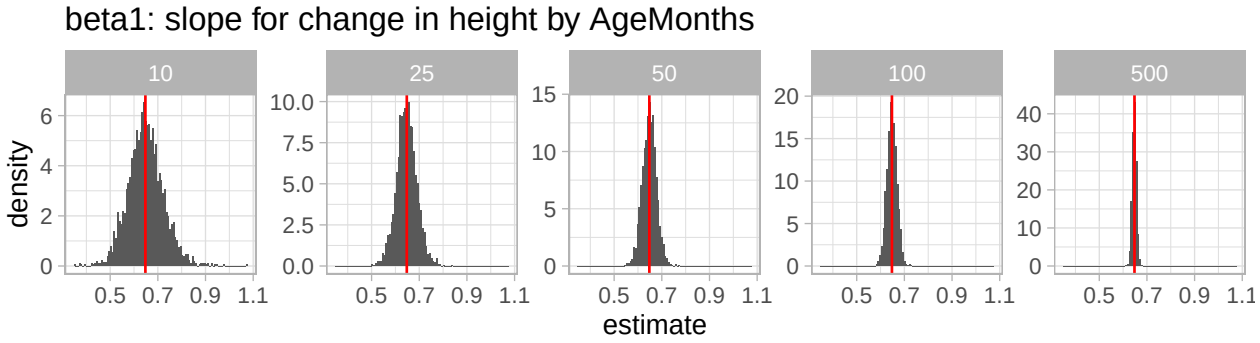
```
#' @param n simulated sample size.
#' @param data a data frame from which to subsample with replacement.
#' @param formula a formula with a numeric outcome and covariate (y ~ x).
#' @param beta1_true the true value for the slope.
#'
#' @return
#' A numeric vector consisting of three elements:
#'   1. The estimated regression slope from the simulated dataset.
#'   2. A binary outcome (0/1) whether the least squares 95% CI contains
#'      the true slope.
#'   3. A binary outcome (0/1) whether the robust 95% CI contains the true
#'      slope.
#'
sim_lm <- function(n, data, formula, beta1_true) {

  df_sim <- data[sample.int(nrow(data), n, replace = TRUE), ]
  fit <- lm(formula, df_sim)

  ci_ls <- confint(fit)[2,]

  covers <- c(coef(fit)[2],
              ci_ls[1] < beta1_true & beta1_true < ci_ls[2])
  names(covers) <- c("estimate", "least_squares")
  covers
}

nsim <- 3000
data <- nhanes_child
formula <- height_all ~ AgeMonths
b1_true <- coef(lm(height_all ~ AgeMonths, nhanes_child))[2]

sim_n10 <- replicate(nsim, sim_lm(10, data, formula, b1_true))
```

```r
sim_n25 <- replicate(nsim, sim_lm(25, data, formula, b1_true))
sim_n50 <- replicate(nsim, sim_lm(50, data, formula, b1_true))
sim_n100 <- replicate(nsim, sim_lm(100, data, formula, b1_true))
sim_n500 <- replicate(nsim, sim_lm(500, data, formula, b1_true))

ex3a <- bind_rows(
  data.frame(n = 10, t(sim_n10)),
  data.frame(n = 25, t(sim_n25)),
  data.frame(n = 50, t(sim_n50)),
  data.frame(n = 100, t(sim_n100)),
  data.frame(n = 500, t(sim_n500))
)
```

b. For each simulated dataset, fit a linear regression model for the height outcome constructed in exercise 2(b) and AgeMonths as the linear predictor (the same regression model estimated in B2(b)). Extract the $\hat{\beta}_1$ coefficient estimate for the slope associated with age. For each sample size, plot a histogram of the distribution of $\hat{\beta}_1$ estimates and compare them to the true best population value (the coefficient estimated in exercise B2(b)). What do you observe about the distribution of $\hat{\beta}_1$.

The distribution of $\hat{\beta}_1$ converges to become closer to the true value of β_1 as the sample size increases.

```r
ex3a %>%
  ggplot(aes(estimate)) +
  geom_histogram(aes(y = ..density..), bins = 100) +
  geom_vline(xintercept = b1_true, color = "red") +
  facet_wrap(~n, scales = "free_y", nrow = 1) +
  theme_light() +
  ggtitle("beta1: slope for change in height by AgeMonths")
```



c. For the regression model fitted to each simulated dataset, extract the 95% confidence interval using the `confint()` function. For each simulated sample size, calculate the proportion of 95% confidence intervals that contain the true value of β_1. How does this change with sample size relative to the nominal 95% coverage target?

For the least-squares 95% CI, the coverage is around 90-91% for all simulated sample sizes. The coverage of the 95% CI does not converge to the theoretical 95% level as the sample size increases due to the violations of the regression assumptions identified in question B2.

```r
ex3a %>%
  mutate(n = as.factor(n)) %>%
  group_by(n) %>%
  summarise(lsq_coverage = mean(least_squares)) %>%
  ggplot(aes(n, lsq_coverage)) +
  geom_col() +
  geom_hline(yintercept = 0.95, linetype = "dashed") +
  coord_cartesian(ylim = c(0.8, 1.0)) +
  theme_light() +
  labs(title = "Coverage of least-squares 95% CI",
       y = "95% CI coverage probability")
```