Problem 4

HW1

Hanyuan Chi(chixx105), Zhi Shen(shenx704)

April 10, 2017

```
suppressPackageStartupMessages({
  library(purrr)
  library(broom)
  library(tidyr)
  library(ggplot2)
  library(dplyr)
})

set.seed(12345)
```

Monte-Carlo Simulation for Autocorrelation

Question 1

Please simulate 1 sample containing $N = 100L$ observations for the following linear regression model

$$y = 0.2 + 0.5 \cdot x + \varepsilon$$

where ε is an AR(1) process with auto-correlation $\text{Corr}(\varepsilon_t, \varepsilon_{t-1}) = 0.75$

- Hint:
  - Assume `X` is picked uniformly randomly in $[-1, 1]$ interval
  - You can use `arima.sim()` method here to simulate ε.
- Output:
  - Please create a *data.frame* `df1` that contains numeric vector `df1$X` contains the generated `X` variable, `df1$Y` contains the generated dependent variables and `df1$e` contains the generated disturbances.

```
N <- 100L
set.seed(12345)

# These are true population coefficients
b0 <- 0.2
b1 <- 0.5

df1 <- data.frame(X = runif(N,-1,1),
                  e = arima.sim(list(order = c(1,0,0), ar = 0.75), n = N))%>%
  mutate(Y = b0 + b1*X + e)
```

Question 2

Please use regular OLS model to estimate the coefficients b from that sample. Please report these coefficients as well as the standard error estimates and 95% confidence interval.

Also, please demonstrate the presence of autocorrelation with a plot!

- Hint:
  - Use `lm()` for linear model and `summary()` for display purposes

```
# Please write your code below

lm_model <- lm(Y~X, data = df1)

summary(lm_model)
```

```
##
## Call:
## lm(formula = Y ~ X, data = df1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6056 -1.2521 -0.1603  1.0989  4.3030
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.5998     0.1717   3.494 0.000717 ***
## X             0.4861     0.2867   1.695 0.093171 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.716 on 98 degrees of freedom
## Multiple R-squared:  0.0285, Adjusted R-squared:  0.01858
## F-statistic: 2.874 on 1 and 98 DF,  p-value: 0.09317
```

```
confint(lm_model)
```

```
##                   2.5 %    97.5 %
## (Intercept)  0.25909250 0.9404826
## X           -0.08288199 1.0551695
```

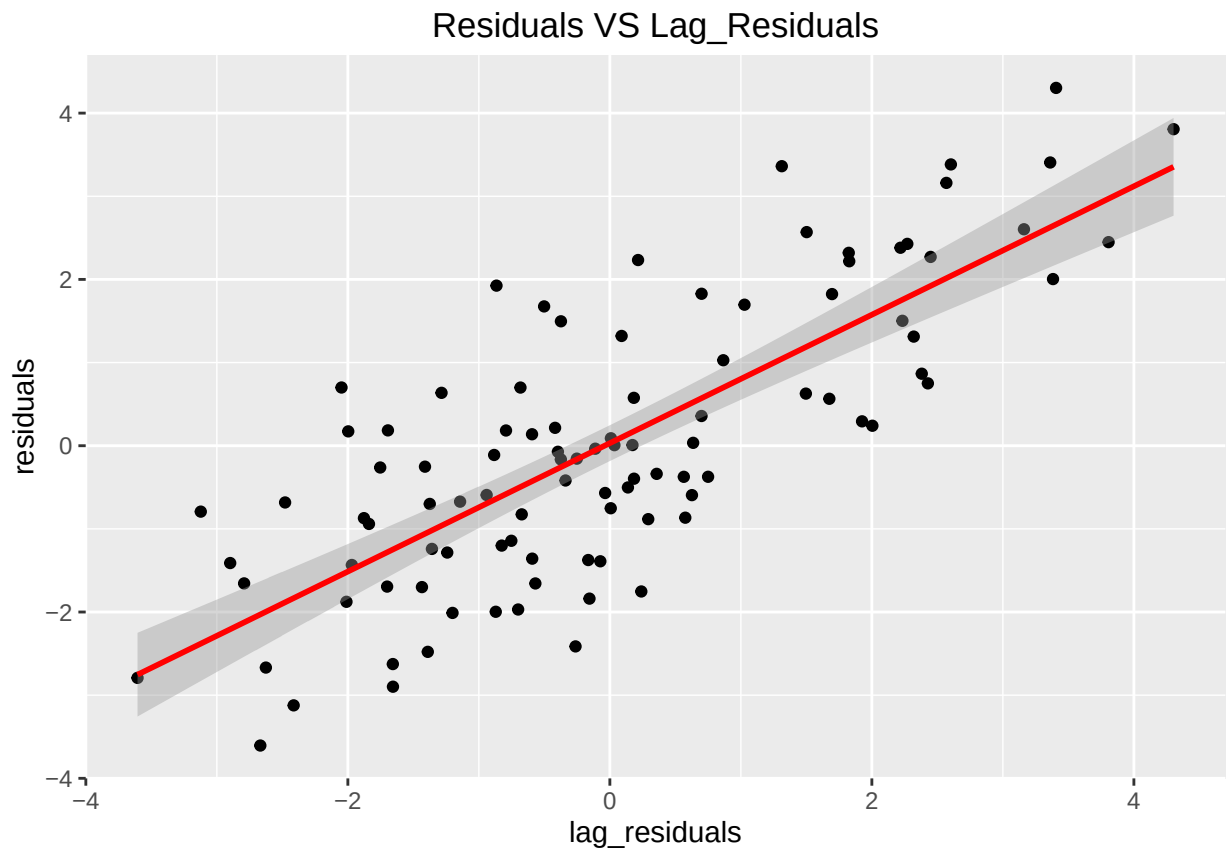
```
tidy(lm_model,conf.int = TRUE)
```

```
##          term  estimate std.error statistic      p.value    conf.low
## 1 (Intercept) 0.5997876 0.1716808  3.493619 0.0007168278  0.25909250
## 2           X 0.4861437 0.2867398  1.695418 0.0931712153 -0.08288199
##   conf.high
## 1 0.9404826
## 2 1.0551695
```

```
ggplot(df1,aes(lag(residuals(lm_model)),residuals(lm_model))) +
  geom_point() +
  stat_smooth(method = "lm", col = "red") +
  ggtitle('Residuals VS Lag_Residuals') +
  ylab('residuals') +
  xlab('lag_residuals')
```

```
## Warning: Removed 1 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```



Question 3

Please revise your code from Question 1 to generate $R = 2000L$ independent samples with $N = 100L$ observations each

- Hint:
  - Try to avoid using loops. Use `dplyr`.
  - Think very carefully about which elements you need to resample and which elements you *do not need* to resample
  - To answer the above, please remember the actual assumptions of an ordinary linear regression
- Output:
  - Please create a *data.frame* `df3` that contains numeric vector `df3$X` contains the generated `X` variable, `df3$Y` contains the generated dependent variables and `df3$e` contains the generated disturbances, `df3$id` contains the id of the sample

```
set.seed(12345)

R <- 2000L

# Please write your code below
df3 <- data.frame(X = rep(runif(N,-1,1), times = R),
                  id = rep(1:R, each = N),
                  e = arima.sim(list(order = c(1,0,0), ar = 0.75), n = N*R))%>%
  mutate(Y = b0 + b1*X + e)
```

Question 4

Please revise your code from Question 2 to estimate R coefficients b from each of those samples. This implies that you should generate a set of R coefficient estimates.

- Hint:
  - Go for long format instead of wide format when necessary.
  - Try to avoid using loops. Use `tidyr` and `nest()`. Also, you may want to use `purrr::map()` and `broom::tidy()`.

```
# Please write your code below

df4 <- df3 %>%
  group_by(id) %>%
  nest() %>%
  mutate(estimated_model = map(data, ~lm(Y~X, data =.))) %>%
  mutate(estimated_coef = map(estimated_model, ~tidy(., conf.int = TRUE))) %>%
  unnest(estimated_coef)

head(df4)
```

```
## # A tibble: 6 x 8
##      id        term  estimate std.error statistic      p.value    conf.low
##   <int>       <chr>     <dbl>     <dbl>     <dbl>        <dbl>       <dbl>
## 1     1 (Intercept) 0.5997876 0.1716808  3.493619 7.168278e-04  0.25909250
## 2     1           X 0.4861437 0.2867398  1.695418 9.317122e-02 -0.08288199
## 3     2 (Intercept) 0.5235069 0.1245789  4.202211 5.837424e-05  0.27628413
## 4     2           X 0.5236944 0.2080706  2.516908 1.346155e-02  0.11078511
## 5     3 (Intercept) 0.5552439 0.1516369  3.661667 4.062418e-04  0.25432532
## 6     3           X 0.5233993 0.2532626  2.066627 4.140647e-02  0.02080799
## # ... with 1 more variables: conf.high <dbl>
```

Question 5

Please plot the histograms of coefficient estimates `b0` and `b1` against the true value

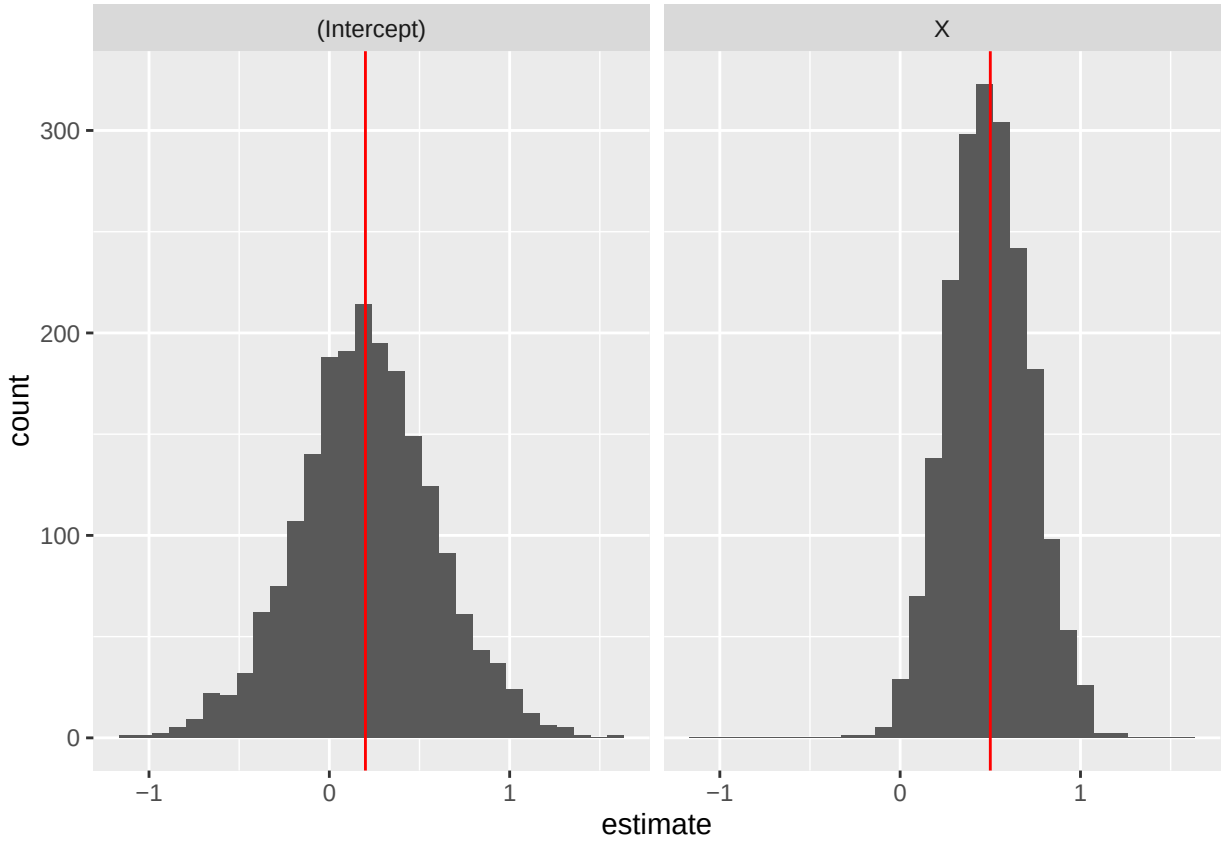
- Hint:
  - Please use `ggplot()`
  - Use `geom_histogram()` to plot the histogram
  - Use `geom_vline(..., color = "red")` to display the true mean
  - Use `facet_grid()` to display them side by side
- Answer the following questions:
  - Is the estimation of true value indeed unbiased?

ANS: Yes, the estimators for both true values are indeed unbiased because both histograms are distributed symmetrically around the true values.

```
# Please write your code below
true_df <- data.frame(term = c("(Intercept)","X"),
                      true_value = c(b0, b1),
                      stringsAsFactors = FALSE
                      )
p5 <- ggplot(df4) +
  geom_histogram(aes(estimate), bins=30) +
  geom_vline(aes(xintercept = true_value),
             color = "red",
```

```
            data = true_df) +
  facet_grid(~term)

p5
```



Question 6

Please estimate the true standard deviation of coefficients `b0` and `b1` and compare it to the estimate you obtained in Question 2.

- Answer the following questions:
  - Did Question 2 produce a good estimate of true variability across different samples?

ANS: The standard errors of both the intercept and the X coefficient in Q2 are not good estimates of the true variability across different samples.

```
# Please write your code below
df4 %>%
  group_by(term) %>%
  summarise(mean(estimate), sd(estimate))
```

```
## # A tibble: 2 x 3
##          term mean(estimate) sd(estimate)
##         <chr>          <dbl>        <dbl>
## 1 (Intercept)      0.2086145    0.3777005
## 2           X      0.4921763    0.2216568
```

Question 7

Please count how often the 95% confidence interval contains true value for each `b0` and `b1` (separately)

- Hints:
  - Join with true values first, then count
- Answer the following questions:
  - Did 95% confidence interval contain the true value in approximately 95% of cases?

ANS: Even though for this seed(12345), the 95% confidence interval for the X coefficient indeed contains the true value for approximately 95% of the samples, but after trying other seeds, the 95% CIs for X coefficient don't always contain the true value for approximately 95% of the samples. So overall, we conclude that the 95% CIs for both the intercept and the X coefficient don't always contain the true value for approximately 95% of the samples and in particular, the 95% CI for the intercept is way overconfident.

```
# Please write your code below
df4 %>%
  inner_join(true_df, by="term") %>%
  group_by(term) %>%
  mutate(contains = ifelse(conf.low <= true_value &
                             true_value <= conf.high,
                           1L,0L)) %>%
  summarise(mean(contains))
```

```
## # A tibble: 2 x 2
##          term mean(contains)
##         <chr>          <dbl>
## 1 (Intercept)         0.5745
## 2           X         0.9715
```

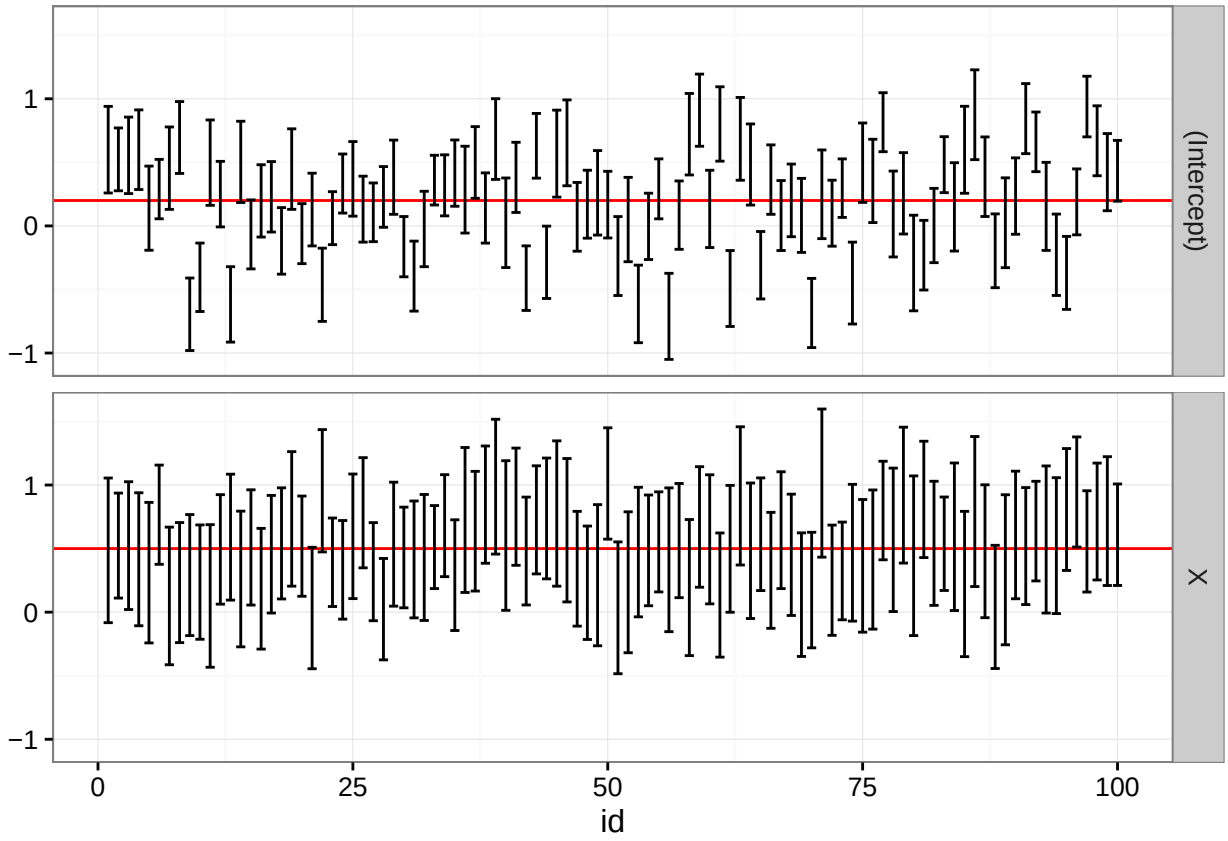
Question 8

Please plot the first 100 of confidence intervals for both `b0` and `b1`, also please plot the true values

- Hints:
  - Use `geom_errorbar(aes(x=...,ymin=...,ymax=...))` for confidence intervals
  - Use `geom_hline(...)` for true values
  - Use `facet_grid()` for vertical positioning instead of horizontal

```
# Please write your code below
p8 <- ggplot(df4 %>% filter(id<=100)) +
  geom_hline(aes(yintercept = true_value),
             color = "red",
             data = true_df) +
  geom_errorbar(aes(x=id, ymin=conf.low, ymax=conf.high)) +
  facet_grid(term~.) +
  theme_bw()

p8
```



Question 9

Please write down a short summary of the results.

- What kind of a problem will you experience in ordinary linear regression estimation if your error terms have some auto-correlation (such as in time series)?

Please be very precise in terms of what is biased and what is not – you need to mention which estimator is biased and which is not biased. You also need to comment on 95% confidence intervals. if you fail to mention some of these, or say, things that are not correct, this will be points off.

(Please do not talk about efficiency of any estimators here as we have no basis to decide it based on these simulations)

> Please write your answer below: In ordinary linear regression estimation, if the error terms have some auto-correlation, 1)the estimates for both the intercept and the X coefficient will stay unbiased 2)but the standard errors of both the intercept and the X coefficient are biased as representatives of the true variability across different samples. 3)As for the 95% confidence intervals, the 95% CIs for both the intercept and the X coefficient don't always contain the true value for approximately 95% of the samples, with 95% CI for the intercept way overconfident.