STATS701 Homework 1

Jordan Farrer

2016-09-25

Contents

Setup

```
set.seed(44)
Sys.setlocale("LC_ALL", "en_US.UTF-8")
```

```
## [1] "en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8"
```

```
options(scipen = 999, digits = 4)

# Create a logger function
logger <- function(msg, level = "info", file = log_file) {
    cat(paste0("[", format(Sys.time(), "%Y-%m-%d %H:%M:%S.%OS"), "][", level, "] ", msg, "\n"), file = 
}

base_dir <- ''
data_dir <- paste0(base_dir, "data/")
code_dir <- paste0(base_dir, "code/")
viz_dir <- paste0(base_dir, "viz/")
```

```
# Source a function that will be used to load/install packages
source('00_fn_load_packages.R')
# Create a vector of packages
packages <- c('dplyr','tidyr','readr','stringr','ggplot2','ggthemes')
# Use function to load the required packages
invisible(lapply(packages, fn_load_packages))
```

```
## [2016-09-19 13:07:50.50][info] Loaded package dplyr version 0.5.0
## [2016-09-19 13:07:50.50][info] Loaded package tidyr version 0.6.0.9000
## [2016-09-19 13:07:50.50][info] Loaded package readr version 1.0.0
## [2016-09-19 13:07:50.50][info] Loaded package stringr version 1.1.0
## [2016-09-19 13:07:50.50][info] Loaded package ggplot2 version 2.1.0
## [2016-09-19 13:07:50.50][info] Loaded package ggthemes version 3.2.0
```

```r
# Create a color palette
pal538 <- ggthemes_data$fivethirtyeight

# Create a theme to use throughout the analysis
theme_jrf <- function(base_size = 12, base_family = "Helvetica") {
    theme(
        plot.background = element_rect(fill = "#F0F0F0", colour = "#606063"),
        panel.background = element_rect(fill = "#F0F0F0", colour = NA),
        panel.border = element_blank(),
        panel.grid.major =   element_line(colour = "#D7D7D8"),
        panel.grid.minor =   element_line(colour = "#D7D7D8", size = 0.25),
        panel.margin =       unit(0.25, "lines"),
        panel.margin.x =     NULL,
        panel.margin.y =     NULL,
        axis.ticks.x = element_blank(),
        axis.ticks.y = element_blank(),
        axis.title = element_text(colour = "#A0A0A3"),
        axis.text.x = element_text(vjust = 1, family = 'Helvetica', colour = '#3C3C3C'),
        axis.text.y = element_text(hjust = 1, family = 'Helvetica', colour = '#3C3C3C'),
        legend.background = element_blank(),
        legend.key = element_blank(),
        plot.title = element_text(face = 'bold', colour = '#3C3C3C', hjust = 0)

    )
}
```

Question 2

Data Loading

Let's use Hadley's readr package to load the dataset, using the col_name parameter to set the column names of the tibble.

```r
survey_results <- read_csv(paste0(data_dir,'Survey_results_final.csv'), skip = 1,
                           col_names = c('hitid','hittypeid','title','description','keywords',
                            'reward','creationtime','maxassignments','requesterannotation',
                            'assignmentdurationinseconds','autoapprovaldelayinseconds',
                            'expiration','numberofsimilarhits','lifetimeinseconds',
                            'assignmentid','workerid','assignmentstatus','accepttime',
                            'submittime','autoapprovaltime','approvaltime','rejectiontime',
                            'requesterfeedback','worktime','lifetimeapprovalrate',
                            'last30daysapprovalrate','last7daysapprovalrate','age',
                            'education','gender','income','sirius','wharton','approve','reject'))

survey_results %>% select(age, education, gender, income, sirius, wharton, worktime)
```

```
## # A tibble: 1,764 × 7
##      age                                  education gender
##    <chr>                                      <chr>  <chr>
## 1     21 Some college, no diploma; or Associate’s degree Female
## 2     56 Some college, no diploma; or Associate’s degree Female
## 3     40                Graduate or professional degree Female
## 4     52                Graduate or professional degree Female
## 5     33         Bachelor’s degree or other 4-year degree   Male
## 6     55 Some college, no diploma; or Associate’s degree   Male
## 7     24 Some college, no diploma; or Associate’s degree Female
## 8     40         Bachelor’s degree or other 4-year degree Female
## 9     35         Bachelor’s degree or other 4-year degree Female
## 10    62 Some college, no diploma; or Associate’s degree Female
## # ... with 1,754 more rows, and 4 more variables: income <chr>,
## #   sirius <chr>, wharton <chr>, worktime <int>
```

```
# Put into a new tibble we'll use for cleaning (there will be a final later)
survey_results_cleaning <- survey_results
```

Data Cleaning

We’ll sequentially clean each of the primary variables of the dataset and create exploratory summaries.

Age

Let’s quickly summarize the age variable, noting that it is a character.

```
survey_results_cleaning %>% group_by(age) %>% summarise(cnt = n()) %>% arrange(age)
```

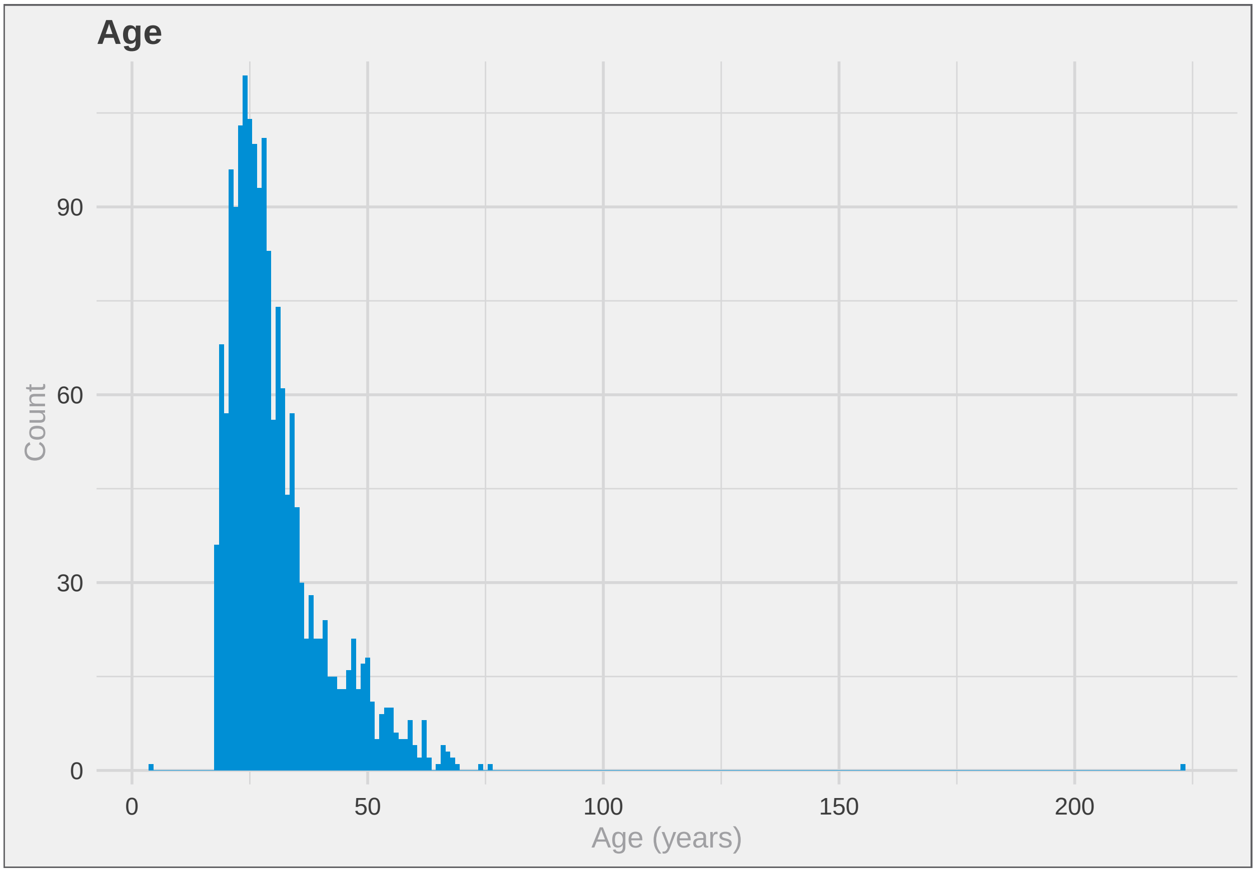
```
## # A tibble: 59 × 2
##      age   cnt
##    <chr> <int>
## 1     18    35
## 2     19    68
## 3     20    57
## 4     21    96
## 5     22    90
## 6    223     1
## 7     23   103
## 8     24   111
## 9     25   104
## 10    26   100
## # ... with 49 more rows
```

We correct some errant values, using our judgement as **data analysts** and plot a histogram.

```
survey_results_cleaning <-
    survey_results %>%
    mutate(
        age2 = ifelse(age == 'Eighteen (18)', "18", ifelse(age == 'female', NA, ifelse(age == "27`", "2
        , age2 = as.integer(age2)
```

```
    )

ggplot(survey_results_cleaning, aes(x = age2)) + geom_histogram(binwidth = 1, fill = pal538['blue']) +
    theme_jrf() +
    scale_x_continuous(expand = c(0.05, 0.01)) + scale_y_continuous(expand = c(0.02, 0.01)) +
    labs(title = "Age", y = "Count", x = "Age (years)")
```



It looks like we still missed some bad values.

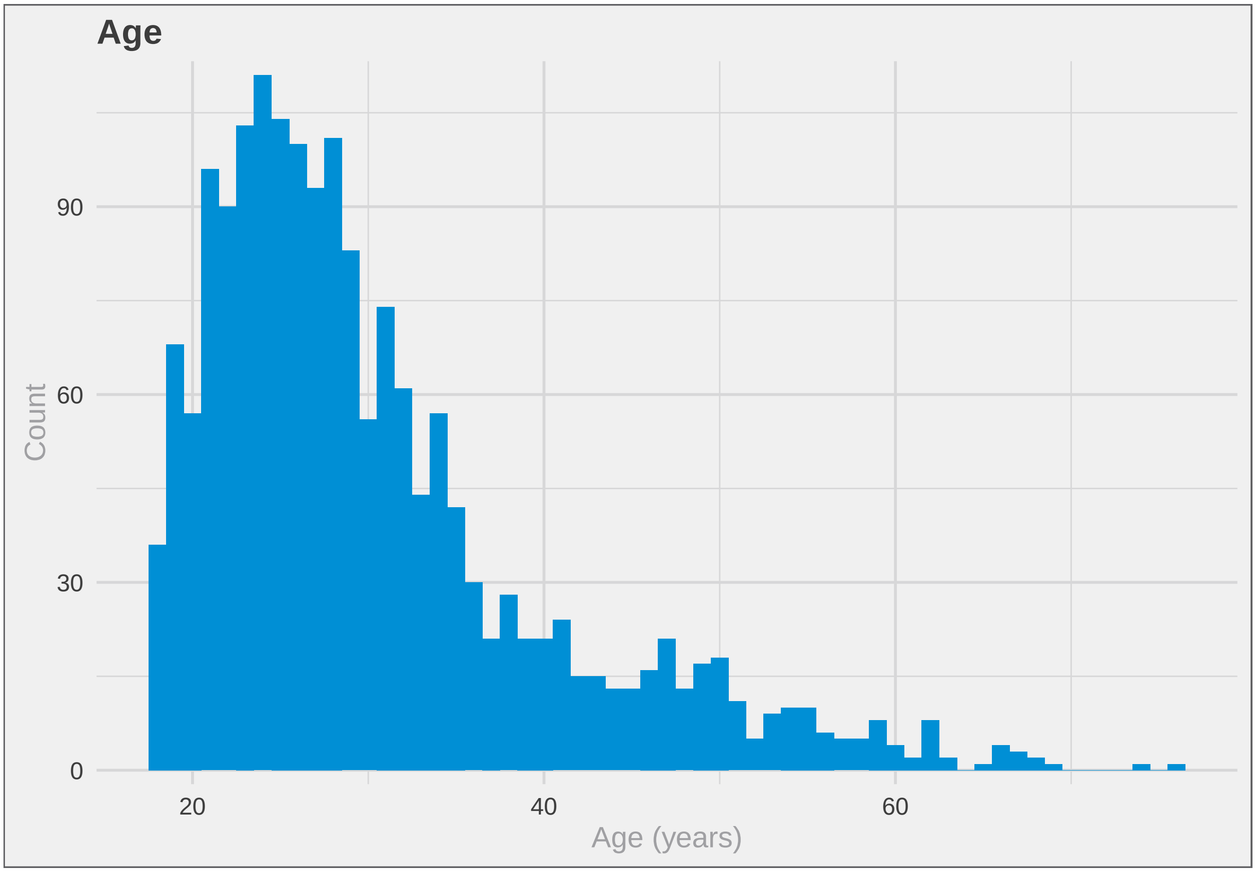
```
sort(unique(survey_results_cleaning$age2))
```

```
##  [1]   4  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33
## [18]  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50
## [35]  51  52  53  54  55  56  57  58  59  60  61  62  63  65  66  67  68
## [52]  69  74  76 223
```

We fix those too and plot the histogram.

```
survey_results_cleaning <-
    survey_results_cleaning %>%
    mutate(
        age3 = ifelse(age2 %in% c(4, 223), NA, age2)
    )
```

```r
ggplot(survey_results_cleaning, aes(x = age3)) + geom_histogram(binwidth = 1, fill = pal538['blue']) +
    theme_jrf() +
    scale_x_continuous(expand = c(0.05, 0.01)) + scale_y_continuous(expand = c(0.02, 0.01)) +
    labs(title = "Age", y = "Count", x = "Age (years)")
```



Education

Let's look at the unique values and counts.

```r
survey_results_cleaning %>% group_by(education) %>% summarise(cnt = n()) %>% arrange(education)
```

```
## # A tibble: 7 × 2
##                                          education   cnt
##                                              <chr> <int>
## 1        Bachelor's degree or other 4-year degree   614
## 2                 Graduate or professional degree   181
## 3            High school graduate (or equivalent)   193
## 4      Less than 12 years; no high school diploma    10
## 5                                            Other     2
## 6                                       select one    19
## 7 Some college, no diploma; or Associate's degree   745
```

It appears that 19 respondents left the survey on the default which read 'select one'. We'll update that to 'Other' and modify this variable to be a factor.

```
survey_results_cleaning <-
    survey_results_cleaning %>%
    mutate(
        education2 = ifelse(education == "select one", "Other", education)
        , education2 = factor(education2, levels = c('Less than 12 years; no high school diploma'
                                                        , 'High school graduate (or equivalent)'
                                                        , 'Some college, no diploma; or Associate's deg
                                                        , 'Bachelor's degree or other 4-year degree'
                                                        , 'Graduate or professional degree'
                                                        , 'Other'))
    )

survey_results_cleaning %>% group_by(education2) %>% summarise(cnt = n()) %>% arrange(education2)
```

```
## # A tibble: 6 × 2
##                                        education2   cnt
##                                            <fctr> <int>
## 1      Less than 12 years; no high school diploma    10
## 2            High school graduate (or equivalent)   193
## 3 Some college, no diploma; or Associate's degree   745
## 4        Bachelor's degree or other 4-year degree   614
## 5                 Graduate or professional degree   181
## 6                                           Other    21
```

Gender

We'll summarise the gender variable.

```
survey_results_cleaning %>% group_by(gender) %>% summarise(cnt = n()) %>% arrange(gender)
```

```
## # A tibble: 3 × 2
##   gender   cnt
##    <chr> <int>
## 1 Female   745
## 2   Male  1013
## 3   <NA>     6
```

We update this to be a factor.

```
survey_results_cleaning <-
    survey_results_cleaning %>%
    mutate(gender2 = as.factor(gender))

survey_results_cleaning %>% group_by(gender2) %>% summarise(cnt = n()) %>% arrange(gender2)
```

```
## # A tibble: 3 × 2
##   gender2   cnt
##    <fctr> <int>
## 1  Female   745
## 2    Male  1013
## 3      NA     6
```

Income

```
survey_results_cleaning %>% group_by(income) %>% summarise(cnt = n()) %>% arrange(income)
```

```
## # A tibble: 7 × 2
##              income   cnt
##               <chr> <int>
## 1  $15,000 - $30,000   367
## 2  $30,000 - $50,000   429
## 3  $50,000 - $75,000   377
## 4 $75,000 - $150,000   329
## 5     Above $150,000    47
## 6  Less than $15,000   209
## 7               <NA>     6
```

Let's convert this to a factor variable.

```
survey_results_cleaning <-
    survey_results_cleaning %>%
    mutate(
        income2 = factor(income, levels = c('Less than $15,000'
                                            , '$15,000 - $30,000'
                                            , '$30,000 - $50,000'
                                            , '$50,000 - $75,000'
                                            , '$75,000 - $150,000'
                                            , 'Above $150,000'))
    )

survey_results_cleaning %>% group_by(income2) %>% summarise(cnt = n()) %>% arrange(income2)
```

```
## # A tibble: 7 × 2
##             income2   cnt
##              <fctr> <int>
## 1  Less than $15,000   209
## 2  $15,000 - $30,000   367
## 3  $30,000 - $50,000   429
## 4  $50,000 - $75,000   377
## 5 $75,000 - $150,000   329
## 6     Above $150,000    47
## 7                 NA     6
```

Sirius and Wharton

```
survey_results_cleaning %>% group_by(sirius) %>% summarise(cnt = n()) %>% arrange(sirius)
```

```
## # A tibble: 3 × 2
##   sirius   cnt
##    <chr> <int>
## 1     No   399
## 2    Yes  1360
## 3   <NA>     5
```

```
survey_results_cleaning %>% group_by(wharton) %>% summarise(cnt = n()) %>% arrange(wharton)
```

```
## # A tibble: 3 × 2
##   wharton   cnt
##     <chr> <int>
## 1      No  1690
## 2     Yes    70
## 3    <NA>     4
```

Let's conver these to booleans for better analysis capabilities.

```
survey_results_cleaning <-
    survey_results_cleaning %>%
    mutate(
        sirius2 = ifelse(sirius == "Yes", TRUE, ifelse(sirius == "No", FALSE, NA))
        , wharton2 = ifelse(wharton == "Yes", TRUE, ifelse(wharton == "No", FALSE, NA))
    )

survey_results_cleaning %>% group_by(sirius2) %>% summarise(cnt = n()) %>% arrange(sirius2)
```

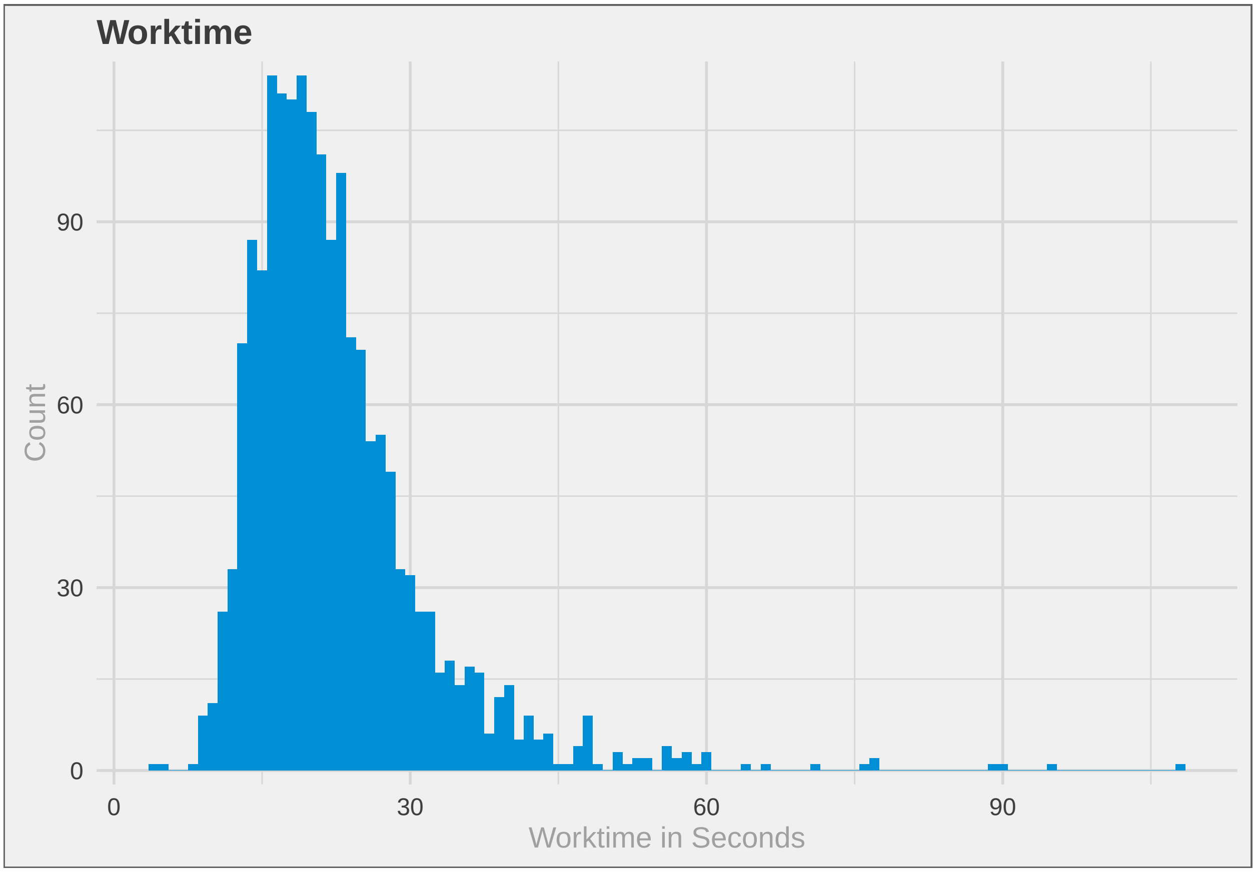
```
## # A tibble: 3 × 2
##   sirius2   cnt
##     <lgl> <int>
## 1   FALSE   399
## 2    TRUE  1360
## 3      NA     5
```

```
survey_results_cleaning %>% group_by(wharton2) %>% summarise(cnt = n()) %>% arrange(wharton2)
```

```
## # A tibble: 3 × 2
##   wharton2   cnt
##      <lgl> <int>
## 1    FALSE  1690
## 2     TRUE    70
## 3       NA     4
```

Worktime

```
ggplot(survey_results_cleaning, aes(x = worktime)) + geom_histogram(binwidth = 1, fill = pal538['blue']]
    theme_jrf() +
    scale_x_continuous(expand = c(0.05, 0.01)) + scale_y_continuous(expand = c(0.02, 0.01)) +
    labs(title = "Worktime", y = "Count", x = "Worktime in Seconds")
```

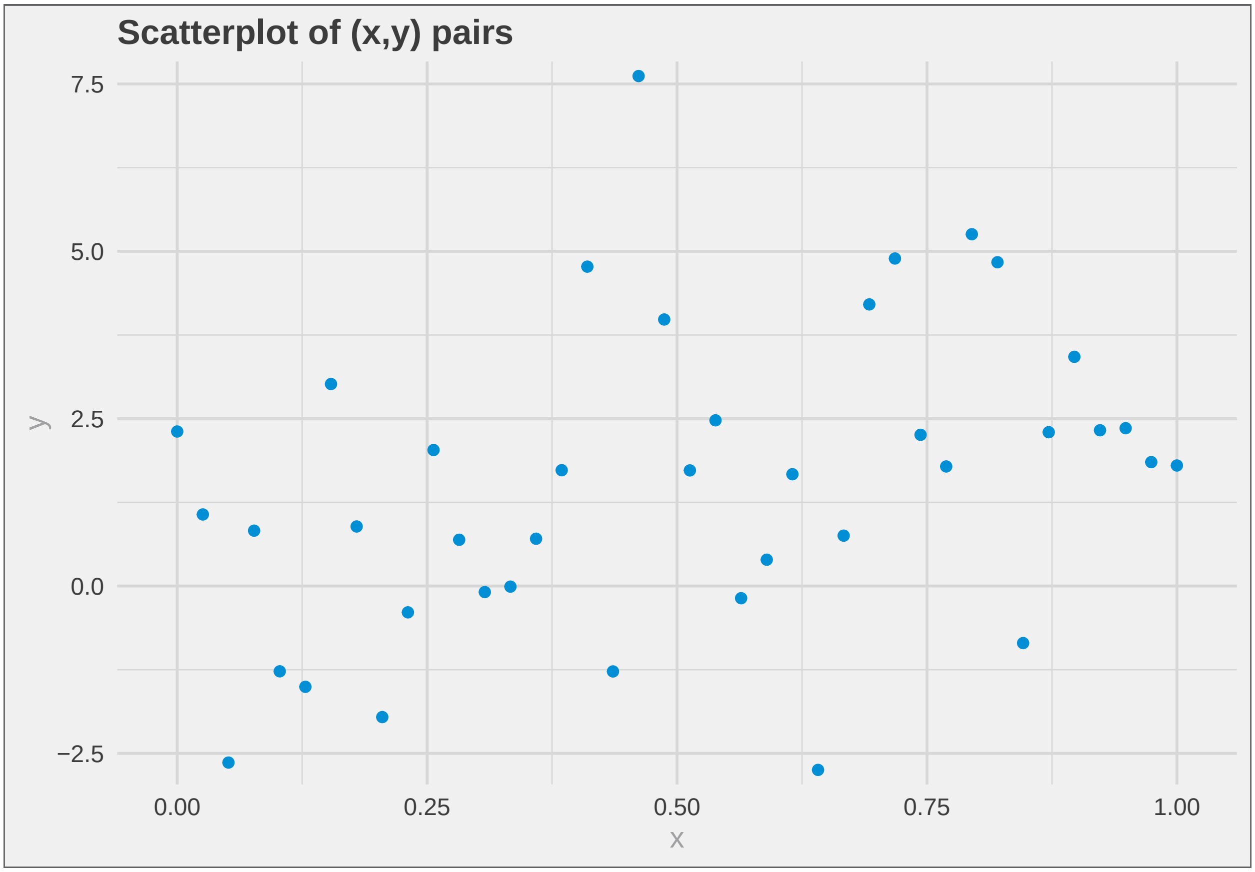


Question 3

Part A

```
x <- seq(0, 1, length = 40)
y <- 1 + 1.2*x + rnorm(40, mean = 0, sd = 2)

ggplot(data_frame(x, y), aes(x = x, y = y)) + geom_point(colour = pal538['blue']) +
    theme_jrf() +
    scale_x_continuous(expand = c(0.05, 0.01)) + scale_y_continuous(expand = c(0.02, 0.01)) +
    labs(title = "Scatterplot of (x,y) pairs", y = "y", x = "x")
```



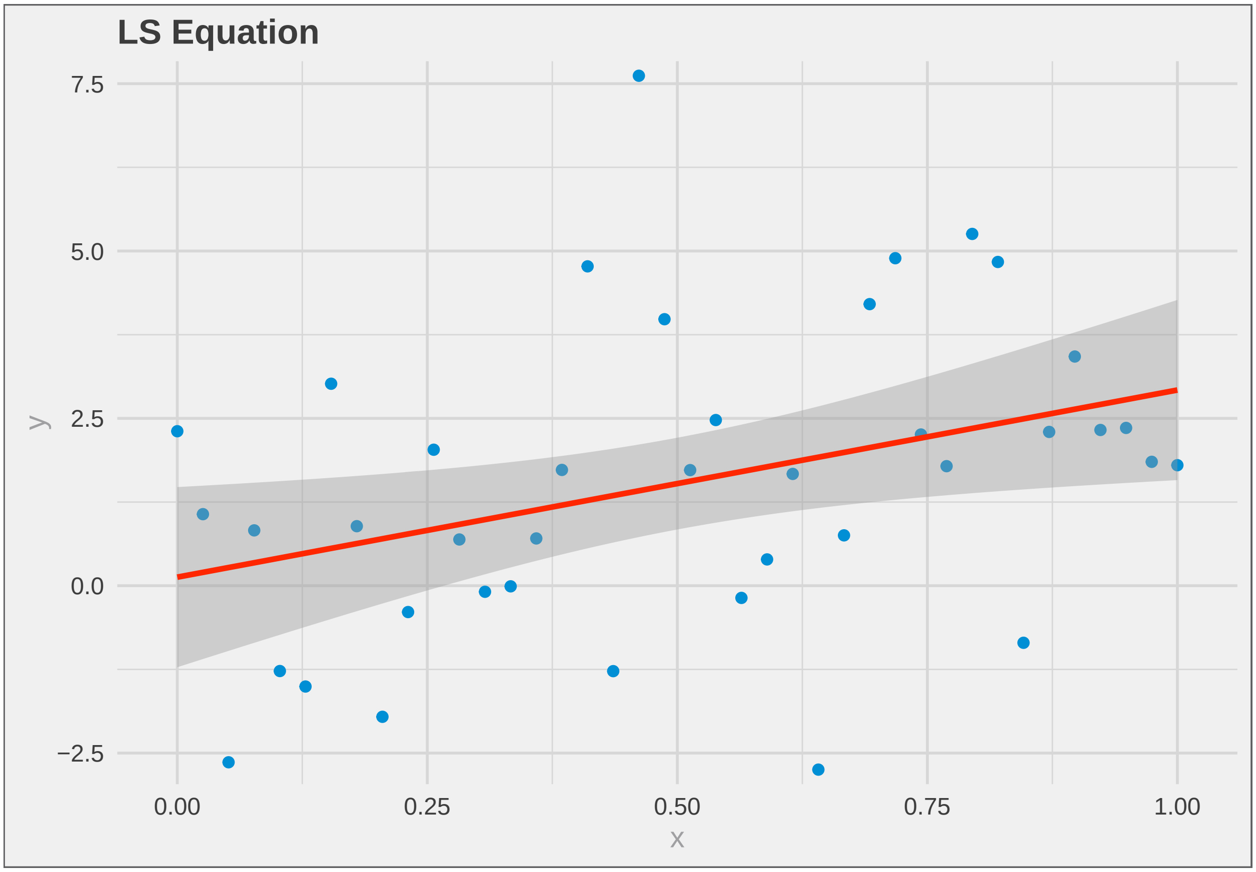
We use the the lm fuction to create a linear model.

```
fit1 <- lm(y ~ x)
summary(fit1)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -4.667 -1.185 -0.246  0.949  6.200
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.128      0.665    0.19    0.848
## x              2.795      1.144    2.44    0.019 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.14 on 38 degrees of freedom
## Multiple R-squared:  0.136,  Adjusted R-squared:  0.113
## F-statistic: 5.97 on 1 and 38 DF,  p-value: 0.0193
```

We find that $\beta_0 = 0.128$ and $\beta_1 = 2.7952$. Next we overlay LS equation on the scatterplot.

```r
ggplot(data = fit1$model, aes(x = x, y = y)) + geom_point(colour = pal538['blue']) +
    geom_smooth(method="lm", se = TRUE, colour = pal538['red']) +
    theme_jrf() +
    scale_x_continuous(expand = c(0.05, 0.01)) + scale_y_continuous(expand = c(0.02, 0.01)) +
    labs(title = "LS Equation", y = "y", x = "x")
```



The 95% confidence interval for β_1 is

$$2.7952 \pm 1.96 \times 1.1441$$

or

$$(0.4791, 5.1112)$$

This 95% confidence interval does indeed contain the true β_1 which is 1.2.

The RSE is 2.1417 which is very close to the true standard deviation of the error of $\sigma = 2$.

Part B

We begin with the given simulation code chunk:

```r
x <- seq(0, 1, length = 40)
n_sim <- 100
b1 <- numeric(n_sim) # nsim many LS estimates of beta1 (=1.2)
upper_ci <- numeric(n_sim) # lower bound
lower_ci <- numeric(n_sim) # upper bound
```

```r
t_star <- qt(0.975, 38)

# Carry out the simulation
for (i in 1:n_sim){
    y <- 1 + 1.2 * x + rnorm(40, sd = 2)
    lse <- lm(y ~ x)
    lse_out <- summary(lse)$coefficients
    se <- lse_out[2, 2]
    b1[i] <- lse_out[2, 1]
    upper_ci[i] <- b1[i] + t_star * se
    lower_ci[i] <- b1[i] - t_star * se
}
```

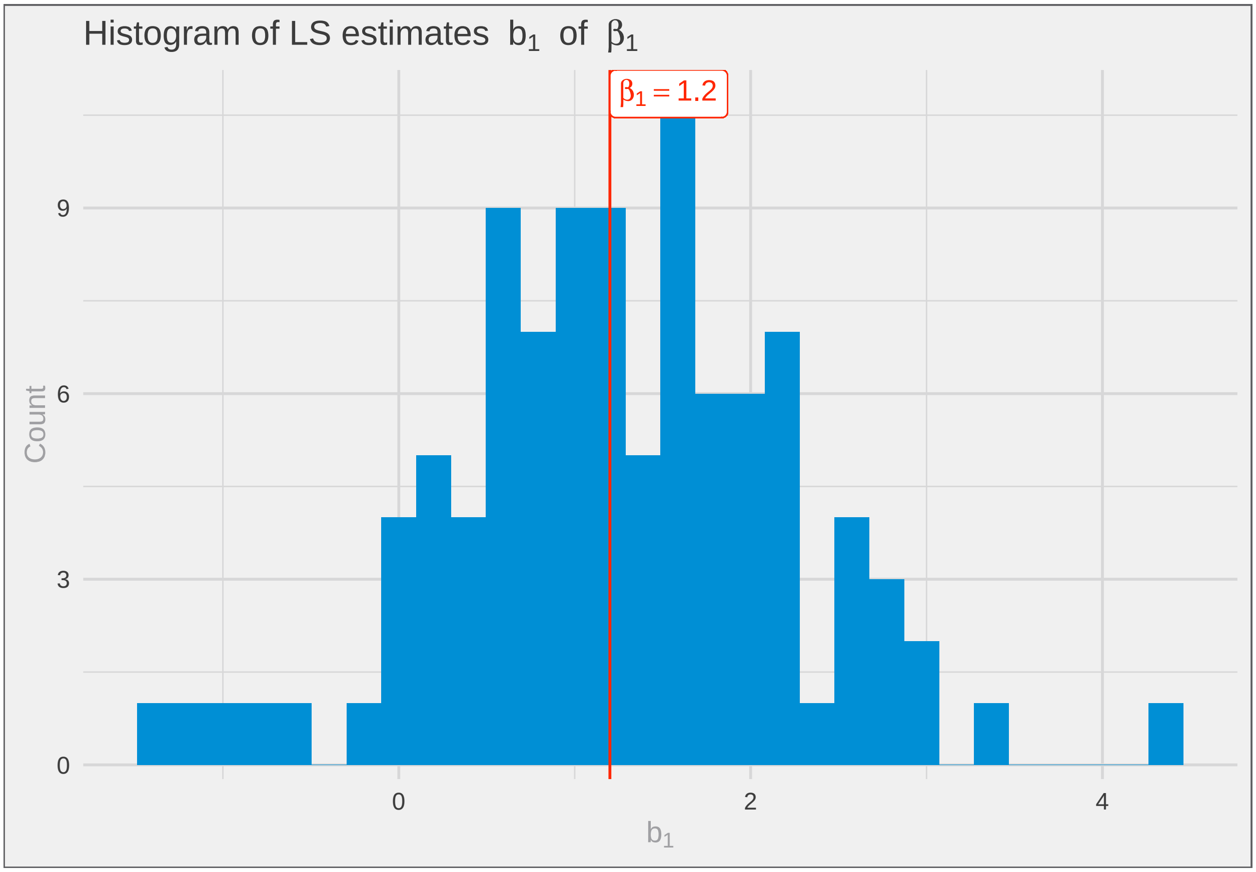
We will summarise β_1.

```r
summary(b1)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   -1.32    0.62    1.23    1.25    1.87    4.42
```

```r
ggplot(data = data_frame(b1 = b1), aes(x = b1)) + geom_histogram(fill = pal538['blue']) +
    geom_vline(xintercept = 1.2, colour = pal538['red']) +
    geom_label(aes(x = 1.2, y = Inf, label = 'beta[1] == 1.2'),
               vjust = "inward", hjust = "inward", parse = TRUE, colour = pal538['red']) +
    theme_jrf() +
    scale_x_continuous(expand = c(0.05, 0.01)) + scale_y_continuous(expand = c(0.02, 0.01)) +
    labs(title = expression("Histogram of LS estimates "~b[1]~" of "~beta[1]), y = "Count", x = express
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

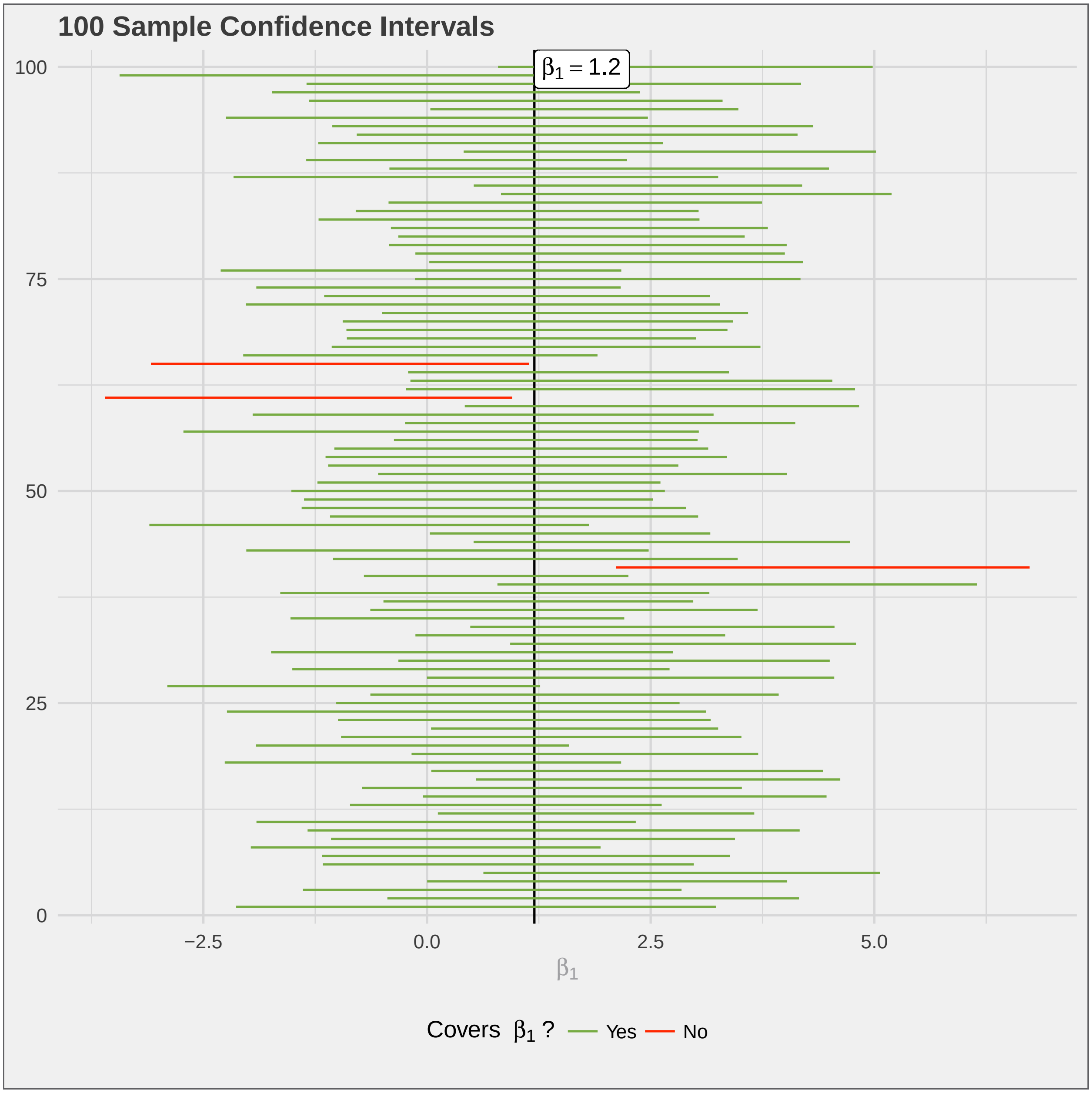


The sampling distribution does agree with the theory as most of the LS estimate of β_1 are close to 1.2.

```
ci <- data_frame(n = 1:100, b1 = b1 , lower_ci = lower_ci, upper_ci = upper_ci,
                 covers = factor(ifelse(lower_ci < 1.2 & upper_ci > 1.2, "Yes", "No"), levels = c("Yes"
```

We find that 97 out of 100 95% confidence intervals cover the true β_1. We show this graphically below, where the red intervals do not cover the true β_1 and the green intervals do cover the true β_1.

```
ggplot(data = ci) +
    geom_vline(xintercept = 1.2) +
    geom_segment(aes(x = lower_ci, xend = upper_ci, y = n, yend = n, colour = covers)) +
    labs(title = "100 Sample Confidence Intervals", y = NULL, x = expression(beta[1])) +
    geom_label(aes(x = 1.2, y = Inf, label = 'beta[1] == 1.2'), vjust = "inward", hjust = "inward", par
    guides(color = guide_legend(title = expression("Covers "~beta[1]~"?"))) +
    theme(legend.position = 'bottom') +
    theme_jrf() +
    scale_x_continuous(expand = c(0.05, 0.01)) + scale_y_continuous(expand = c(0.02, 0.01)) +
    scale_colour_manual(values = c('Yes' = pal538['green'][[1]], 'No' = pal538['red'][[1]]))
```



100 Sample Confidence Intervals
$\beta_1 = 1.2$
100
75
50
25
0
−2.5
0.0
2.5
5.0
β_1
Covers β_1 ? Yes No