Data Taming Assignment 2

Dongju Ma

2024-07-02

Setup

```
# Load the required packages
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(inspectdf)
library(caret)
```

```
## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(moments)
```

Q1. Loading the data

```
# Your student number and calculate file number
ysn = 1942340
filenum <- (ysn+1) %% 3
filenum
```

```
## [1] 0
```

```
# Read in the data
arch0 <- read.csv('./data/archery_0.csv')
head(arch0,10)
```

```
##      Archer  Commenced                                RES
## 1     Kevin 26/07/1999 Target hit 58 times from 68 shots.
## 2     Naomi 14/11/2014 Target hit 34 times from 51 shots.
## 3    Daniel 28/08/2011 Target hit 44 times from 62 shots.
## 4      Kate 22/03/2008 Target hit 52 times from 66 shots.
## 5   Micheal  2/04/2015 Target hit 43 times from 63 shots.
## 6      Leah 17/03/2011 Target hit 38 times from 55 shots.
## 7  Rhiannon 24/10/2021 Target hit 19 times from 52 shots.
## 8      Jake 23/04/2017 Target hit 30 times from 50 shots.
## 9   Krystal 24/05/2013 Target hit 40 times from 62 shots.
## 10   Willow 30/06/2016 Target hit 32 times from 57 shots.
```

Q2. Tidy up the data

Q2.(a) Replace Archers' names with id

```
# Add row numbers and store the new tibble for comparison
arch1 <- mutate(arch0,Archer = (1:555))
arch1 <- rename(arch1,id = Archer)
```

Q2.(b) Create experience days column

```
# calculate the xp days
arch1 <- mutate(arch1,xp = dmy('15-06-2024') - dmy(Commenced))
arch1 <- relocate(arch1,"xp",.before = Commenced)
```

Q2.(c) Seprate the RES data

```
# extract targets numbers and arrows numbers
arch1 <- mutate(arch1,targets = str_match(arch1$RES, 'hit (\\d+) times from (\\d+) shots')[,2])
arch1 <- mutate(arch1,arrows = str_match(arch1$RES, 'hit (\\d+) times from (\\d+) shots')[,3])
arch1 <- mutate(arch1,RES = NULL)
arch1 <- relocate(arch1,"arrows",.before = targets)
head(arch1,10)
```

```
##    id        xp  Commenced arrows targets
## 1   1 9091 days 26/07/1999     68      58
## 2   2 3501 days 14/11/2014     51      34
## 3   3 4675 days 28/08/2011     62      44
## 4   4 5929 days 22/03/2008     66      52
## 5   5 3362 days  2/04/2015     63      43
## 6   6 4839 days 17/03/2011     55      38
## 7   7  965 days 24/10/2021     52      19
## 8   8 2610 days 23/04/2017     50      30
## 9   9 4040 days 24/05/2013     62      40
## 10 10 2907 days 30/06/2016     57      32
```

Q3. Variables Identification

- id: Categorical Ordinal
  The id number just indicates each archer's name and it's represented as integers which could have an implying order.

- xp: Quantitative Discrete
  The xp column are the number of experience days of the archers. They are limited and discrete integers.

- Commenced : Categorical Ordinal
  The commenced column are the commenced dates of the archers. They could be categorized by the same dates and be ordered by the dates.

- arrows: Quantitative Discrete
  They are the numbers of the archers' shots, which could be any integers but no floats.

- targets: Quantitative Discrete
  As the same to the arrows, but they are the numbers of hits.

Q4. Taming data

```
# Change column titles in lower case
arch1 <- rename(arch1,commenced = Commenced)
# Change xp days to integers
arch1$xp <- as.integer(arch1$xp)
# Change commenced dates into year-month-day format
arch1$commenced <- dmy(arch1$commenced)
# Change id and commenced dates to factor type
arch1$id <- as.factor(arch1$id)
arch1$commenced <- as.factor(arch1$commenced)
# Change arrows and targets into integers
arch1$arrows <- as.integer(arch1$arrows)
arch1$targets <- as.integer(arch1$targets)
# Check is there any impossible data in the tibble
inspect_na(arch1)
```

```
## # A tibble: 5 x 3
##   col_name    cnt  pcnt
##   <chr>     <int> <dbl>
## 1 id            0     0
## 2 xp            0     0
## 3 commenced     0     0
## 4 arrows        0     0
## 5 targets       0     0
```

```
# Check the whole tibble for strange numbers
inspect_num(arch1)
```

```
## # A tibble: 3 x 10
##   col_name   min    q1 median   mean    q3   max      sd pcnt_na hist
##   <chr>    <int> <dbl>  <int>  <dbl> <dbl> <int>   <dbl>   <dbl> <named list>
## 1 xp         551 3000.   5425 5496.  7718. 45456 3635.         0 <tibble>
## 2 arrows      50   54      59   59.6   65     70    6.34       0 <tibble>
## 3 targets     16   35      42   42.1   49     64    9.95       0 <tibble>
```

We could see the xp days number with 45456 days is obviously unusual, check the data we can see the commenced dates are set into 1900-01-01. I think we should clean these dates for the next analysis.

```
# Delete the commenced date with too large numbers
arch1 <- filter(arch1,xp < 45456)
# Check the whole tibble for strange numbers again
inspect_num(arch1)
```

```
## # A tibble: 3 x 10
##   col_name   min    q1 median   mean    q3   max      sd pcnt_na hist
##   <chr>    <int> <dbl>  <int>  <dbl> <dbl> <int>   <dbl>   <dbl> <named list>
## 1 xp         551  2989   5423 5351.   7700 10188 2730.         0 <tibble>
## 2 arrows      50    54     59   59.7    65    70    6.35       0 <tibble>
## 3 targets     16    35     43   42.1    49    64    9.93       0 <tibble>
```

```
# Output the tibble
head(arch1,10)
```

```
##    id   xp  commenced arrows targets
## 1   1 9091 1999-07-26     68      58
## 2   2 3501 2014-11-14     51      34
## 3   3 4675 2011-08-28     62      44
## 4   4 5929 2008-03-22     66      52
## 5   5 3362 2015-04-02     63      43
## 6   6 4839 2011-03-17     55      38
## 7   7  965 2021-10-24     52      19
## 8   8 2610 2017-04-23     50      30
## 9   9 4040 2013-05-24     62      40
## 10 10 2907 2016-06-30     57      32
```

Q5. Choose the random sample

```
set.seed(1942340)
arch_sample <- sample_n(arch1,450,replace = FALSE)
head(arch_sample,10)
```

```
##     id   xp  commenced arrows targets
## 1  529 2575 2017-05-28     69      40
## 2  173 7042 2005-03-05     50      43
## 3  175 6549 2006-07-11     64      52
## 4  482 8908 2000-01-25     50      45
## 5  282 6463 2006-10-05     54      40
## 6  120  734 2022-06-12     68      25
## 7  213 3473 2014-12-12     51      30
## 8  254 2077 2018-10-08     70      36
## 9   62 3562 2014-09-14     56      37
## 10 291 3012 2016-03-17     66      41
```

Q6. Calculate the accuracy

```
arch_sample <- mutate(arch_sample,acc = targets/arrows)
head(arch_sample,10)
```

```
##     id   xp  commenced arrows targets       acc
## 1  529 2575 2017-05-28     69      40 0.5797101
## 2  173 7042 2005-03-05     50      43 0.8600000
## 3  175 6549 2006-07-11     64      52 0.8125000
## 4  482 8908 2000-01-25     50      45 0.9000000
## 5  282 6463 2006-10-05     54      40 0.7407407
## 6  120  734 2022-06-12     68      25 0.3676471
## 7  213 3473 2014-12-12     51      30 0.5882353
## 8  254 2077 2018-10-08     70      36 0.5142857
## 9   62 3562 2014-09-14     56      37 0.6607143
## 10 291 3012 2016-03-17     66      41 0.6212121
```

- acc: Quantitative Continuous
  The accuracy number could be any number between 0 and 1.

Q7. Summarize your sample

```
inspect_num(arch_sample)
```

```
## # A tibble: 4 x 10
##   col_name     min       q1   median     mean      q3       max        sd pcnt_na
##   <chr>      <dbl>    <dbl>    <dbl>    <dbl>   <dbl>     <dbl>     <dbl>   <dbl>
## 1 xp       551     3054.    5291     5325.    7711.   10188     2720.          0
## 2 arrows    50       54       59       59.6     66       70        6.44        0
## 3 targets   17       35       42       42.0     49       64        9.69        0
## 4 acc        0.315    0.613    0.729    0.704    0.82     0.967    0.146       0
## # i 1 more variable: hist <named list>
```

The mean for the number of days of experience is 5324.75 days, while the mean for accuracy is 0.70

Q8. Prediction with the standardised variables

```
# preprocess the sample data
arch_preprocess <- preProcess(
  tibble(
  xp = arch_sample$xp,
  acc = arch_sample$acc
))
# predict with the preprocess data
arch_predict <- predict(arch_preprocess,arch_sample)
head(arch_predict,10)
```

```
##     id         xp  commenced arrows targets        acc
## 1  529 -1.0110267 2017-05-28     69      40 -0.8534102
## 2  173  0.6313996 2005-03-05     50      43  1.0643726
## 3  175  0.4501334 2006-07-11     64      52  0.7393709
## 4  482  1.3174904 2000-01-25     50      45  1.3380583
## 5  282  0.4185129 2006-10-05     54      40  0.2483839
## 6  120 -1.6879255 2022-06-12     68      25 -2.3043758
## 7  213 -0.6808501 2014-12-12     51      30 -0.7950799
## 8  254 -1.1941313 2018-10-08     70      36 -1.3010534
## 9   62 -0.6481266 2014-09-14     56      37 -0.2991684
## 10 291 -0.8503506 2016-03-17     66      41 -0.5694478
```

Q9. Standardised data analysis

```
# Show the statistics of standardised variables
standard_data <- tibble(
  xp = arch_predict$xp,
  acc = arch_predict$acc
)
summary(standard_data)
```

```
##        xp                 acc
##  Min.   :-1.75521   Min.   :-2.6659
##  1st Qu.:-0.83509   1st Qu.:-0.6239
##  Median :-0.01241   Median : 0.1714
##  Mean   : 0.00000   Mean   : 0.0000
##  3rd Qu.: 0.87747   3rd Qu.: 0.7907
##  Max.   : 1.78812   Max.   : 1.7979
```

```
inspect_num(standard_data)
```

```
## # A tibble: 2 x 10
##   col_name   min     q1  median      mean    q3   max    sd pcnt_na hist
##   <chr>    <dbl>  <dbl>   <dbl>     <dbl> <dbl> <dbl> <dbl>   <dbl> <named list>
## 1 xp       -1.76 -0.835 -0.0124 -8.54e-17 0.877  1.79     1       0 <tibble>
## 2 acc      -2.67 -0.624  0.171   2.13e-16 0.791  1.80     1       0 <tibble>
```

Q9.(a) mean

The accuracy has a higher mean.

Q9.(b) median

The accuracy has a higher median.

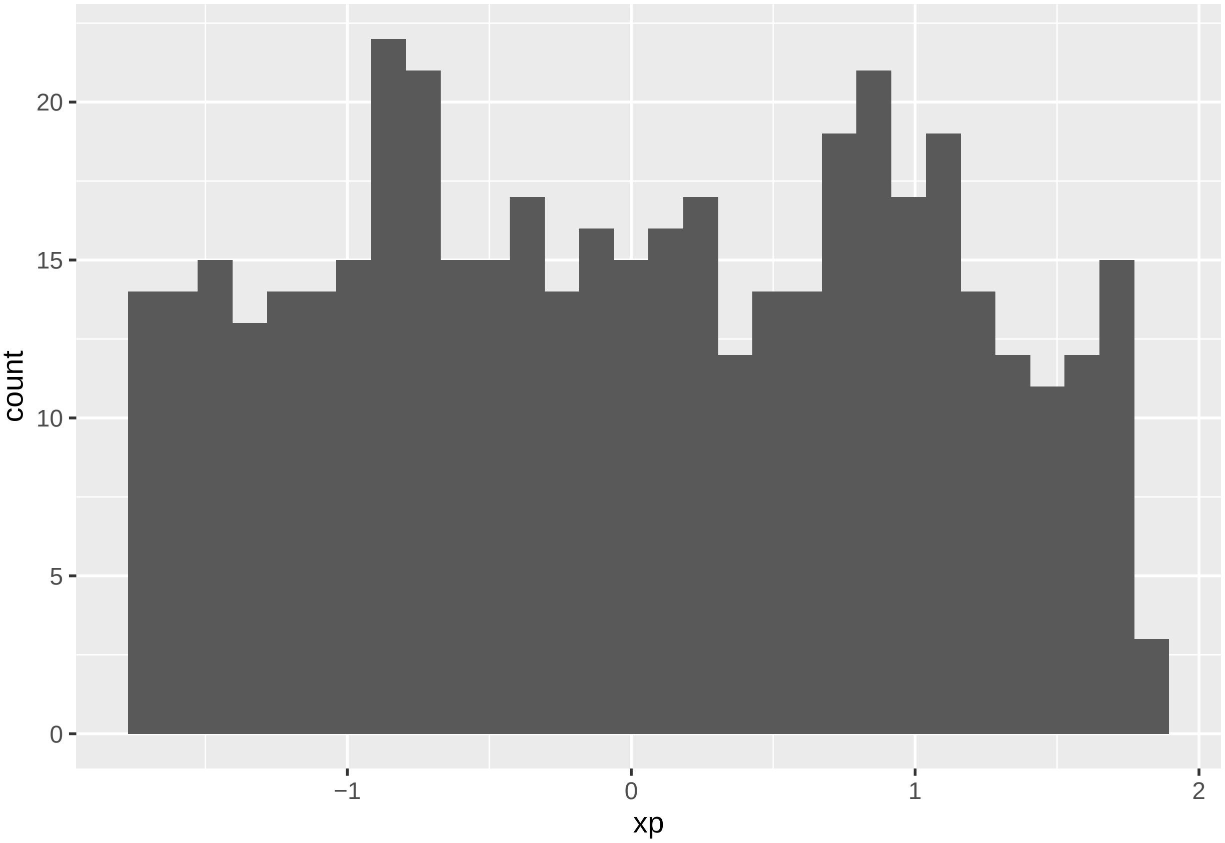
Q9.(c) inter-quartile range

As we calculated, the xp has a higher inter-quartile range.

Q9.(d) Histograms plots and skewness

```
# plot the histograms of each variable
# xp
ggplot(arch_predict,aes(xp)) +
  geom_histogram()
```

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

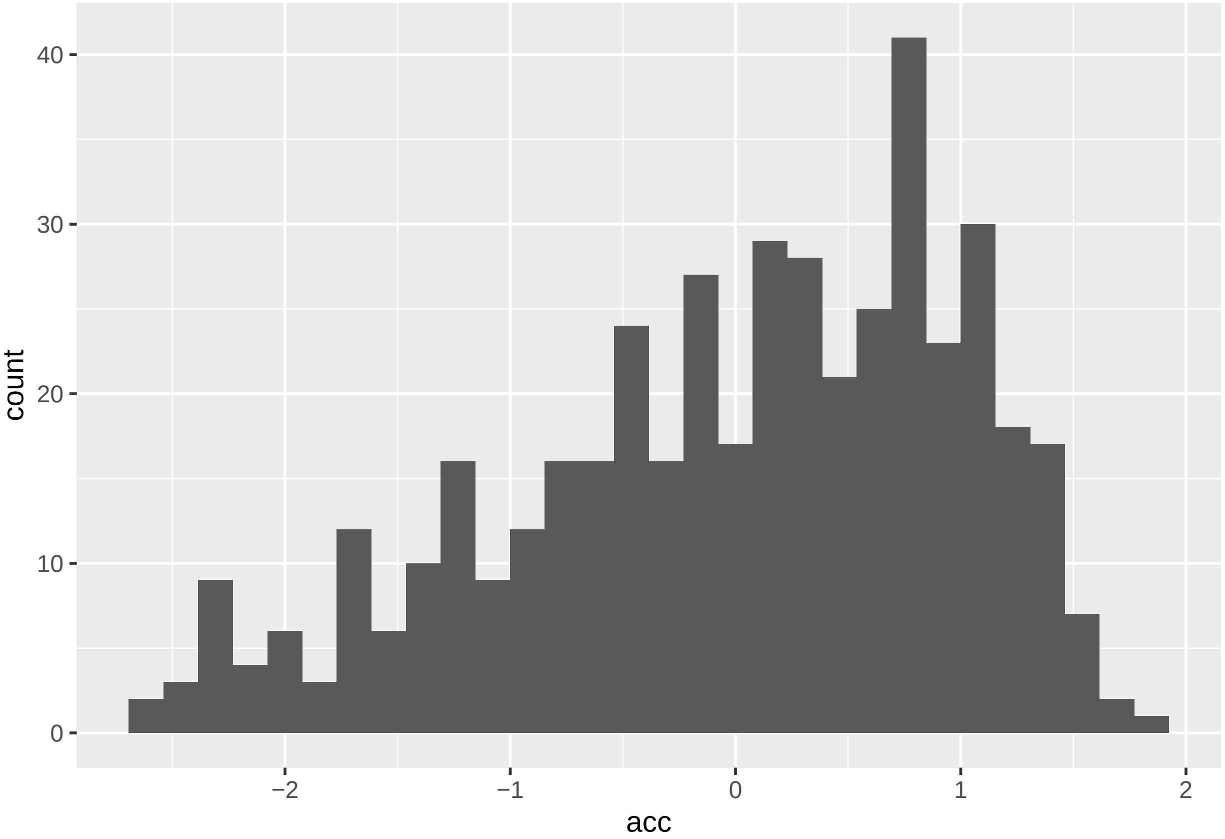


```
skewness(arch_predict$xp)
```

```
## [1] 0.01820871
```

```
#accuracy
ggplot(arch_predict,aes(acc)) +
  geom_histogram()
```

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



```
skewness(arch_predict$acc)
```

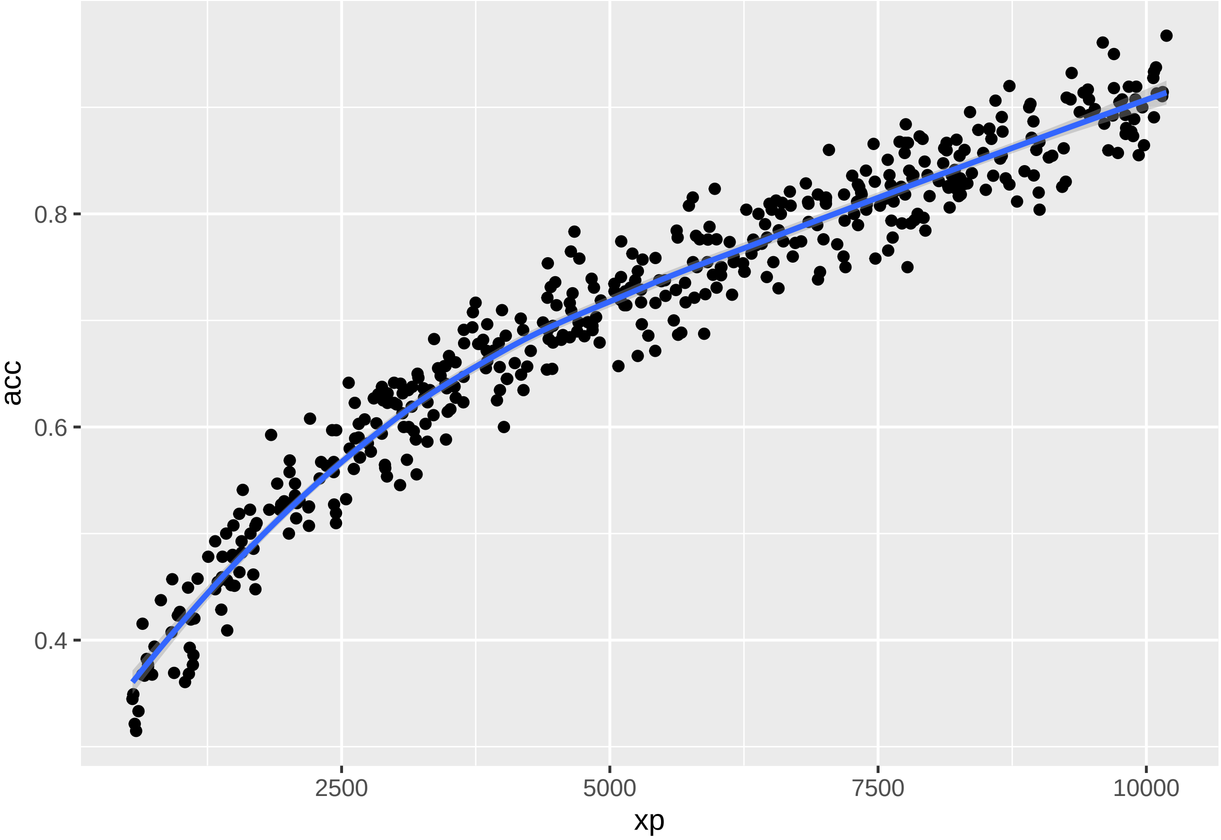
```
## [1] -0.5876215
```

The histogram of xp is very close to symmetric distribution, with the skewness is only 0.0182.
But on the other hand the accuracy's histogram is obviously asymmetric, most bars are on the right side of the histogram, which with the skewness of -0.59 can also make a proof of that.

Q10. Judge the linear relationship by scatterplot

```
# Plot
ggplot(arch_sample,aes(x = xp,y = acc))+
  geom_point()+
  geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



- It doesn't look like a linear relationship because the trend line of the points is not a straight line. It more looks like a curve of $y = logx$.

- The skewness of standardised accuracy is below 0, which means most of the accuracy are higher than median. Meanwhile the distribution of xp number is almost symmetric. So the curve of the scatter point goes flat after the accuracy data reaches median.

Q11 Box-Cox it

Q11.(a) Get the estimate of λ

```
# Use BoxCoxTrans to obtain lambda, with the steps of 0.1 in range [-10,10]
arch_sample_bc <- BoxCoxTrans(
  y = arch_sample$acc,
  x = arch_sample$xp,
```

```r
  lambda = seq(-10,10,0.1)
  )
arch_sample_bc$lambda
```

```
## [1] 2.4
```

The estimate of λ is 2.4.

Q11.(b) Box-Cox the accuracy

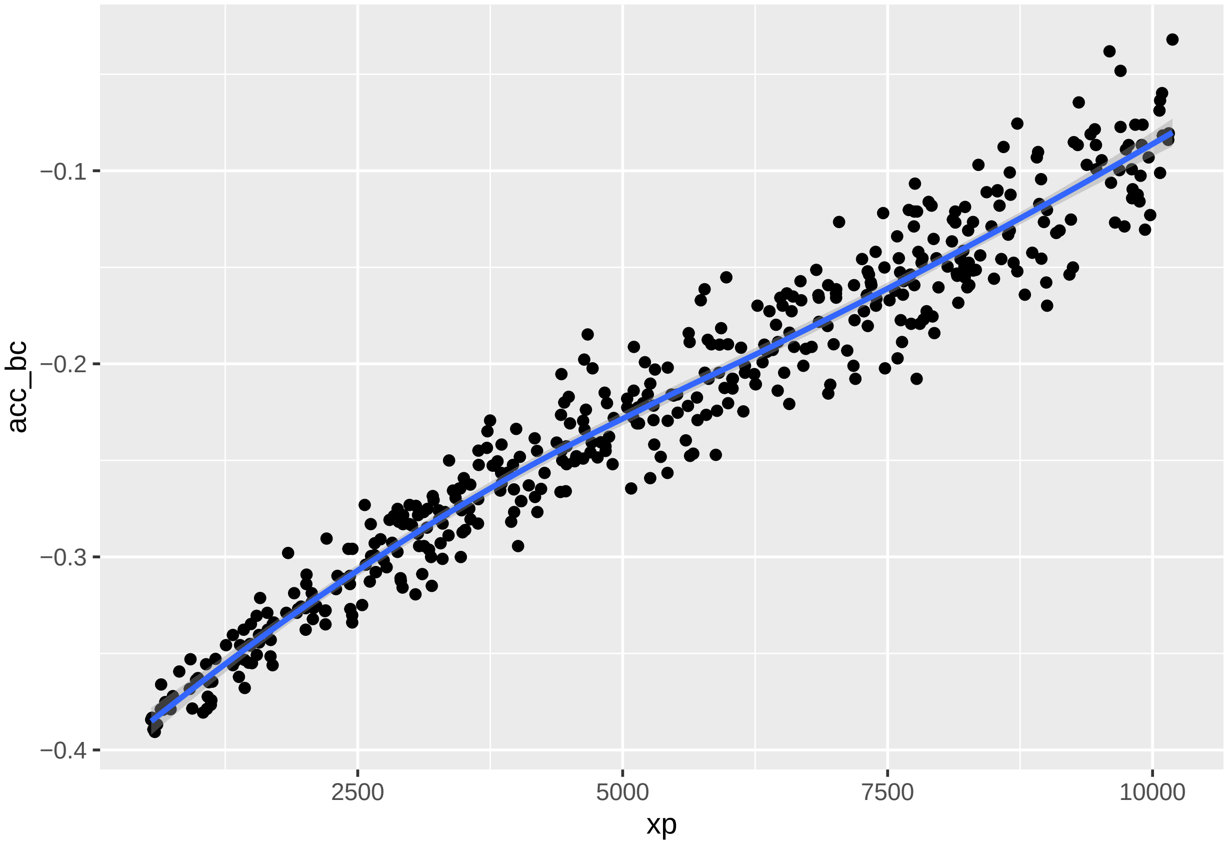
```r
# Predict transformation and add it into a new column
arch_sample<- mutate(arch_sample,acc_bc = predict(arch_sample_bc,arch_sample$acc))
head(arch_sample,10)
```

```
##     id   xp  commenced arrows targets       acc      acc_bc
## 1  529 2575 2017-05-28     69      40 0.5797101 -0.30407794
## 2  173 7042 2005-03-05     50      43 0.8600000 -0.12654174
## 3  175 6549 2006-07-11     64      52 0.8125000 -0.16352429
## 4  482 8908 2000-01-25     50      45 0.9000000 -0.09309478
## 5  282 6463 2006-10-05     54      40 0.7407407 -0.21390413
## 6  120  734 2022-06-12     68      25 0.3676471 -0.37892480
## 7  213 3473 2014-12-12     51      30 0.5882353 -0.30006323
## 8  254 2077 2018-10-08     70      36 0.5142857 -0.33220115
## 9   62 3562 2014-09-14     56      37 0.6607143 -0.26255993
## 10 291 3012 2016-03-17     66      41 0.6212121 -0.28375455
```

Q12 The new scatterplot of Box-Cox transformed data

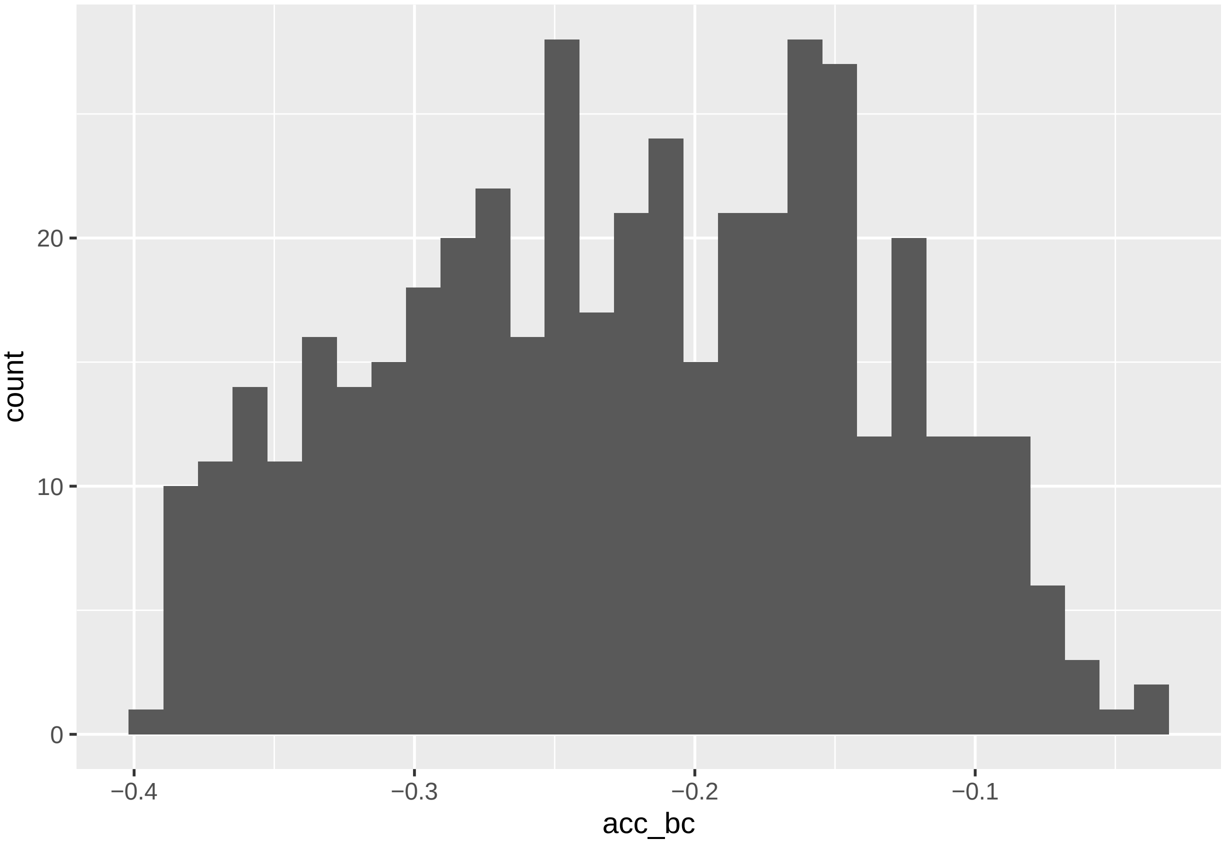
```r
# plot the scatter plot of transformed data against xp
ggplot(arch_sample,aes(x = xp,y = acc_bc))+
  geom_point()+
  geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



```r
# plot the histogram and get skewness
ggplot(arch_sample,aes(acc_bc))+
  geom_histogram()
```

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



```
skewness(arch_sample$acc_bc)
```

```
## [1] -0.05413942
```

- The trend line of scatterplot is straighter than before.
- The skewness is below 0 but closed to 0, and the shape of the histogram is close to a symmetric distribution.
- After the transformation, the relationship becomes much closer to linear than before.

Q13 Linear modeling it

Q13.(a) The eqution of linear model

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

In this equation, y_i is the response variable, in this case is the accuracy.x_i is the independet variable, we could use this to predict y_i. β_0 and β_i are regression coefficients, β_0 is the intercept, β_i is the slope. ϵ_i is the error.

Q13.(b) The equation of this case

$$ACC\_BC_i = \beta_0 + \beta_i XP_i + \epsilon_i$$

In this equation, replace the y_i and x_i with ACC_BC_i and XP_i, which are representing Box-Cox transformed accuracy and experience days.

Q13.(c) Build the linear model with R

```
# Build and assign your linear model
arch_lm <- lm(acc_bc ~ xp,arch_sample)
# Show the statistics
summary(arch_lm)
```

```
##
## Call:
## lm(formula = acc_bc ~ xp, data = arch_sample)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.057067 -0.013937 -0.000767  0.013501  0.059079
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.840e-01  2.061e-03 -186.31   <2e-16 ***
## xp           3.001e-05  3.448e-07   87.04   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01987 on 448 degrees of freedom
## Multiple R-squared:  0.9442, Adjusted R-squared:  0.944
## F-statistic:  7577 on 1 and 448 DF,  p-value: < 2.2e-16
```

```
# Obtain the coefficients
as.numeric(arch_lm$coefficients)
```

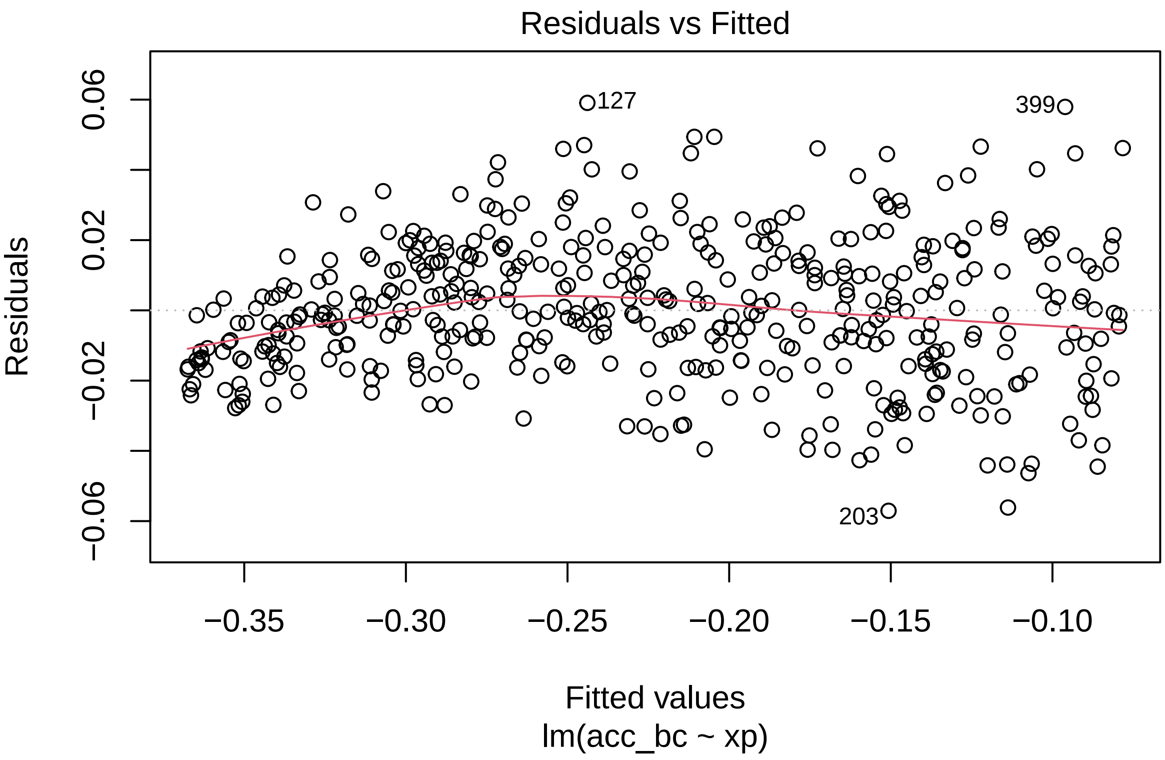
```
## [1] -3.840363e-01  3.001489e-05
```

With the coefficents, we can rewrite the equation as:

$$ACC\_BC_i = -0.38 + 0.00003XP_i + \epsilon_i$$

## Q14 Assumption test - Linearity
We could look at a plot to check it.

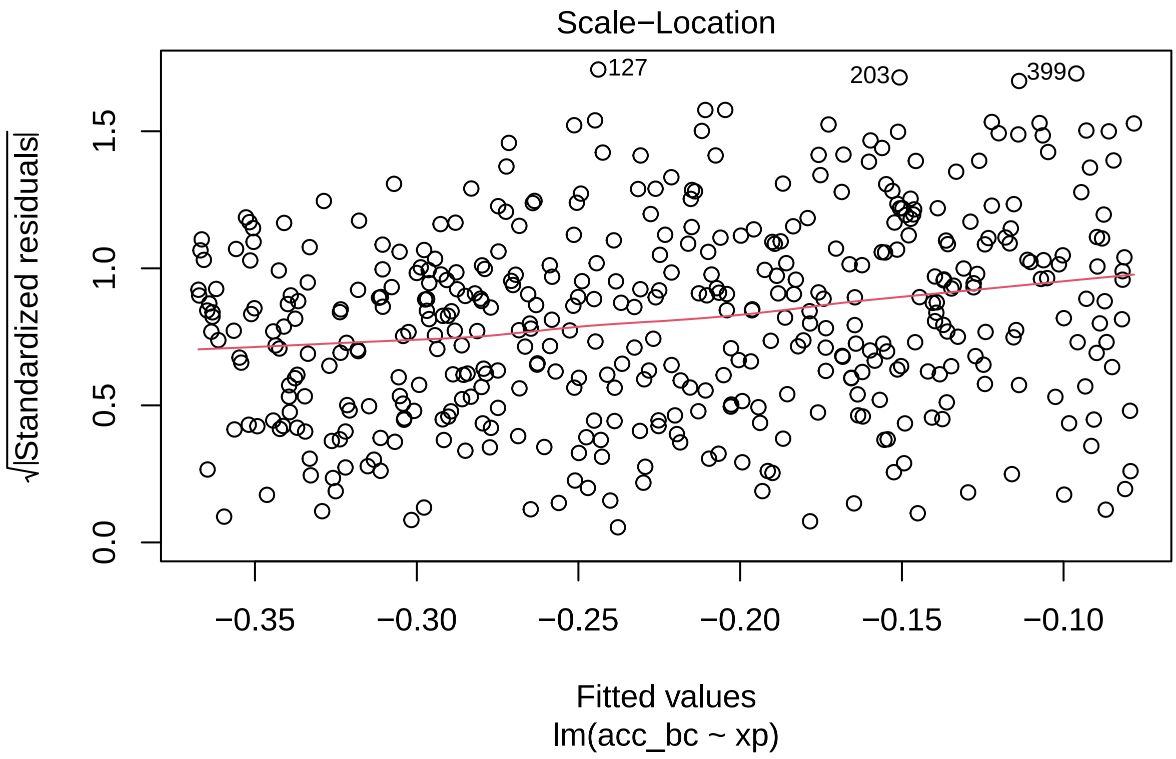
```
plot(arch_lm, which = 1)
```



The red line is roughly straight and the dots are evenly located up and down. So linearity would be a safe assumption.

- Homoscedasticity We could check the assumption by plot, too.

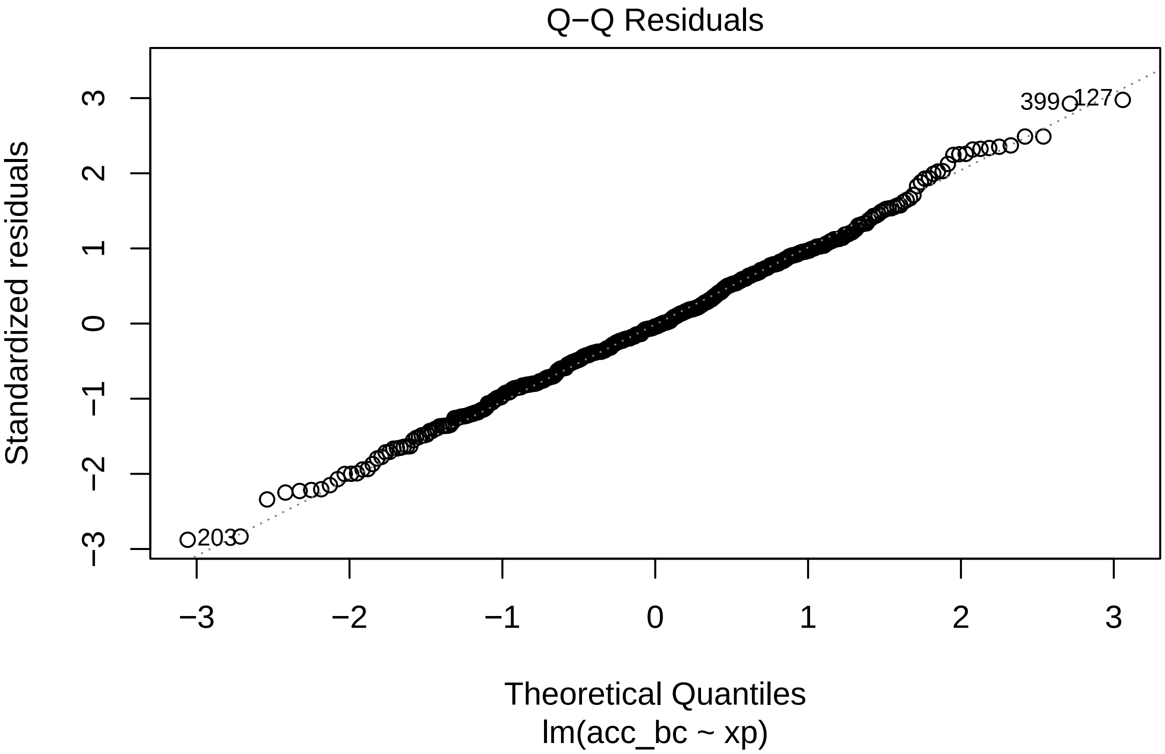
```r
plot(arch_lm, which = 3)
```



The red line is still roughly straight and the dots are evenly spread in vertical direction. This assumption would be safe.This can prove that errors are in a normal distribution which means the errors are independent from each other.

- Normality and independence We still check the assumption by looking at a plot.

```
plot(arch_lm, which = 2)
```



The points are lie along the dotted line and they are concentrating in the center. And there are several dots drifting away from the line which are fine because they are less than -2 on x-axis or greater than +2 on x-axis. Based on these plots, we can assure that this model's linearity and the independence between the independent variables.

Q15 Predict it

```
# Predict with the linear model
pred_values <- tibble(
  xp_years = c(2,5,10,15,20,25),
  xp = xp_years * 365
  )
pred_data <- predict(arch_lm,pred_values,interval = 'confidence',level = 0.99)
pred_values <- mutate(pred_values, fit = pred_data[,1])
pred_values <- mutate(pred_values,lower = pred_data[,2])
pred_values <- mutate(pred_values,upper = pred_data[,3])
# Inverse the box-cox transformed data
pred_values <- mutate(
  pred_values,
  fit = (fit * arch_sample_bc$lambda + 1) ^ ( 1 / arch_sample_bc$lambda)
  )
pred_values <- mutate(
  pred_values,
  lower = (lower * arch_sample_bc$lambda + 1) ^ ( 1 / arch_sample_bc$lambda)
  )
pred_values <-
  mutate(pred_values,
         upper = (upper * arch_sample_bc$lambda + 1) ^ ( 1 / arch_sample_bc$lambda)
         )
pred_values
```

```
## # A tibble: 6 x 5
##   xp_years    xp   fit lower upper
##      <dbl> <dbl> <dbl> <dbl> <dbl>
## 1        2   730 0.429 0.413 0.444
## 2        5  1825 0.522 0.512 0.531
## 3       10  3650 0.639 0.634 0.644
## 4       15  5475 0.732 0.728 0.736
## 5       20  7300 0.811 0.807 0.815
## 6       25  9125 0.880 0.875 0.885
```

Q16 Conclusions

- As far as my observation goes, if you define a good archer is who has the accuracy greater than 0.7, he might need 15 years of practice. And the longer time he spend the less accuracy he could add.
- According to the statistic data before, the median accuracy of the archers is 0.73, and the mean of accuracy is 0.70. Being greater than mean indicates he could beat most archers, but being greater than median could make him reach a better world. So I think a person needs at least 15 years of practice could make him behalves perfect in archery.