Statistical Data Analysis

Afreen Fatima

2023-05-05

```
# Loading libraries
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

Loading dataset

```
dataset <- read.csv("transactions.csv", stringsAsFactors = T)
```

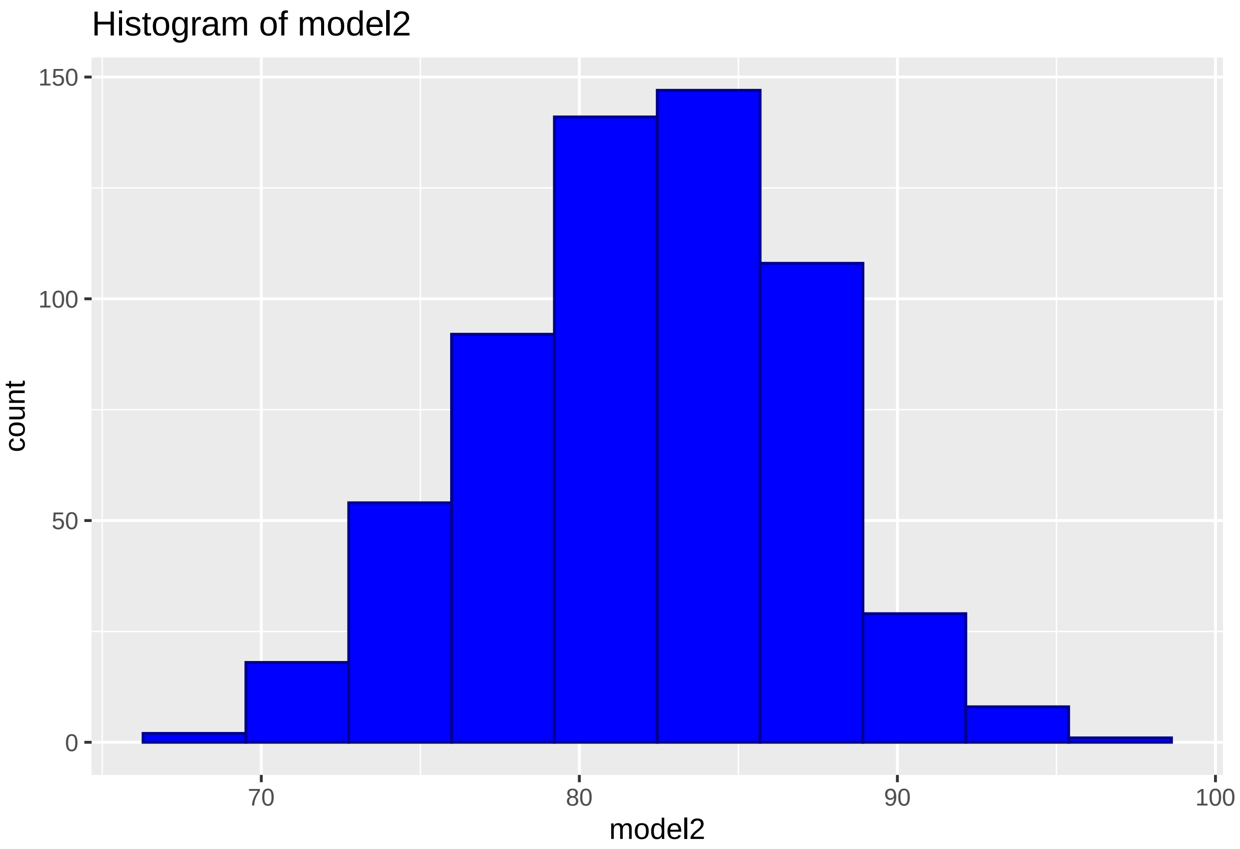
```
# Exploring dataset
View(dataset) ## To see in tabular form
str(dataset)
```

```
## 'data.frame':    600 obs. of  8 variables:
##  $ id       : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ status   : Factor w/ 2 levels "Existing","New": 1 2 1 1 1 2 1 1 1 1 ...
##  $ device   : Factor w/ 3 levels "mobile","PC",..: 3 3 1 1 1 2 1 3 1 1 ...
##  $ recipient: Factor w/ 4 levels "bill","purchase",..: 2 4 3 2 2 2 2 1 2 2 ...
##  $ value    : num  23.79 0.88 250 44.94 30.13 ...
##  $ time     : int  166 196 79 152 139 233 106 221 140 168 ...
##  $ model1   : num  82.5 81.1 80.7 79.2 77.4 75.6 76.4 78.3 75.9 78.2 ...
##  $ model2   : num  84.8 76.7 81.3 86.6 79.4 79.2 76.5 83 85.3 80.7 ...
```

SECTION 1: VISUALISATIONS AND SUMMARY STATISTICS

1.1: Model 2 shows a normal distribution on histogram as it is not skewed on completely one side. Mean and median are calculated using 'mean' and 'median' functions where mean of model 2 is 82 and median is 82.3. Hence we can say the model 2 is normally distributed.

```
# Histogram of model2
ggplot(data = dataset, aes(x = model2)) + geom_histogram(bins = 10, color = "darkblue", fill = "blue") +
labs(title = "Histogram of model2")
```



```
# Applying mean and median function to model2 column
mean(dataset$model2)
```

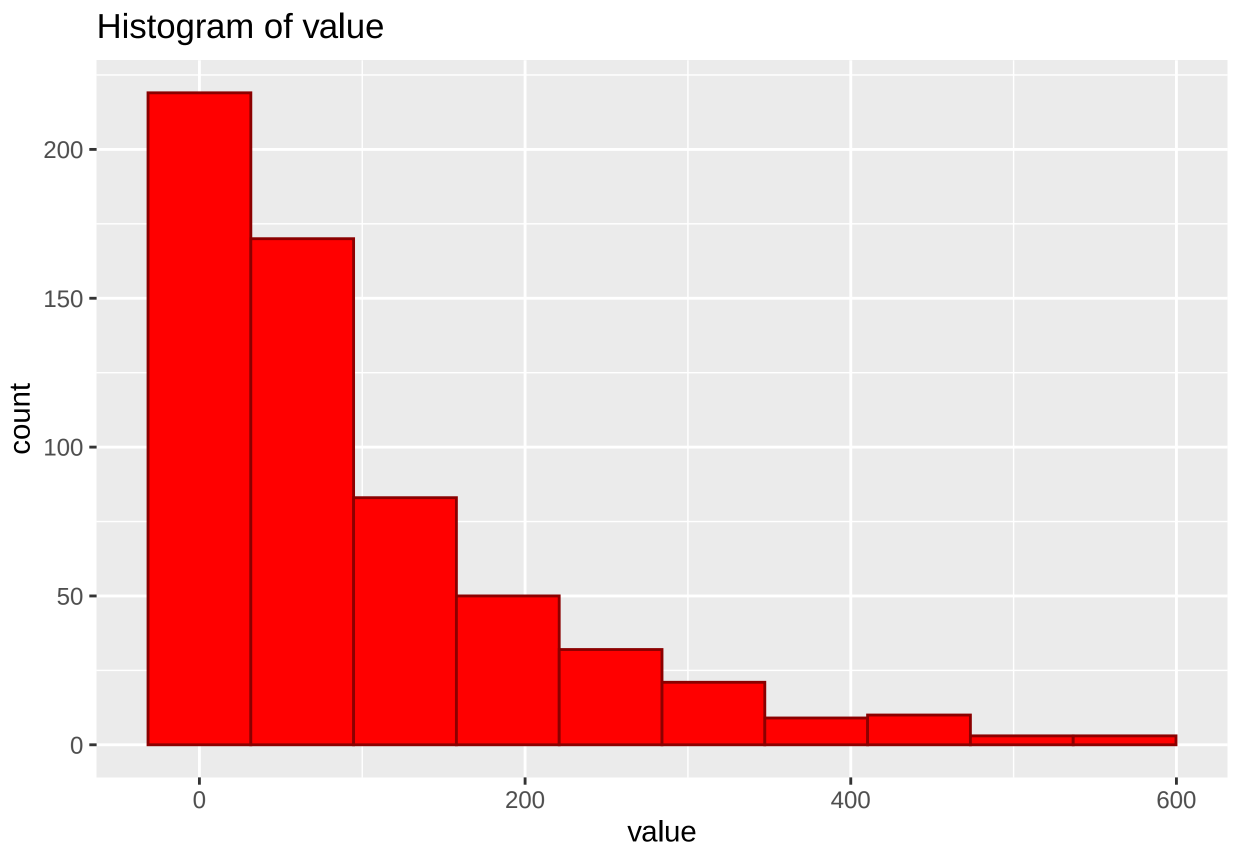
```
## [1] 82.07133
```

```
median(dataset$model2)
```

```
## [1] 82.3
```

1.2 Histogram ‘value’ does not shows a normal distribution as it is a skewed completely one side. Mean and median is calculated using ‘mean’ and ‘median’ functions where mean of value is 96 and median is 52. Hence we can say the column value is NOT normally distributed.

```
# Histogram of value
ggplot(data = dataset, aes(x = value)) + geom_histogram(bins = 10, color = "darkred", fill = "red") +
labs(title = "Histogram of value")
```



```
# Applying mean and median function to value column
mean(dataset$value)
```

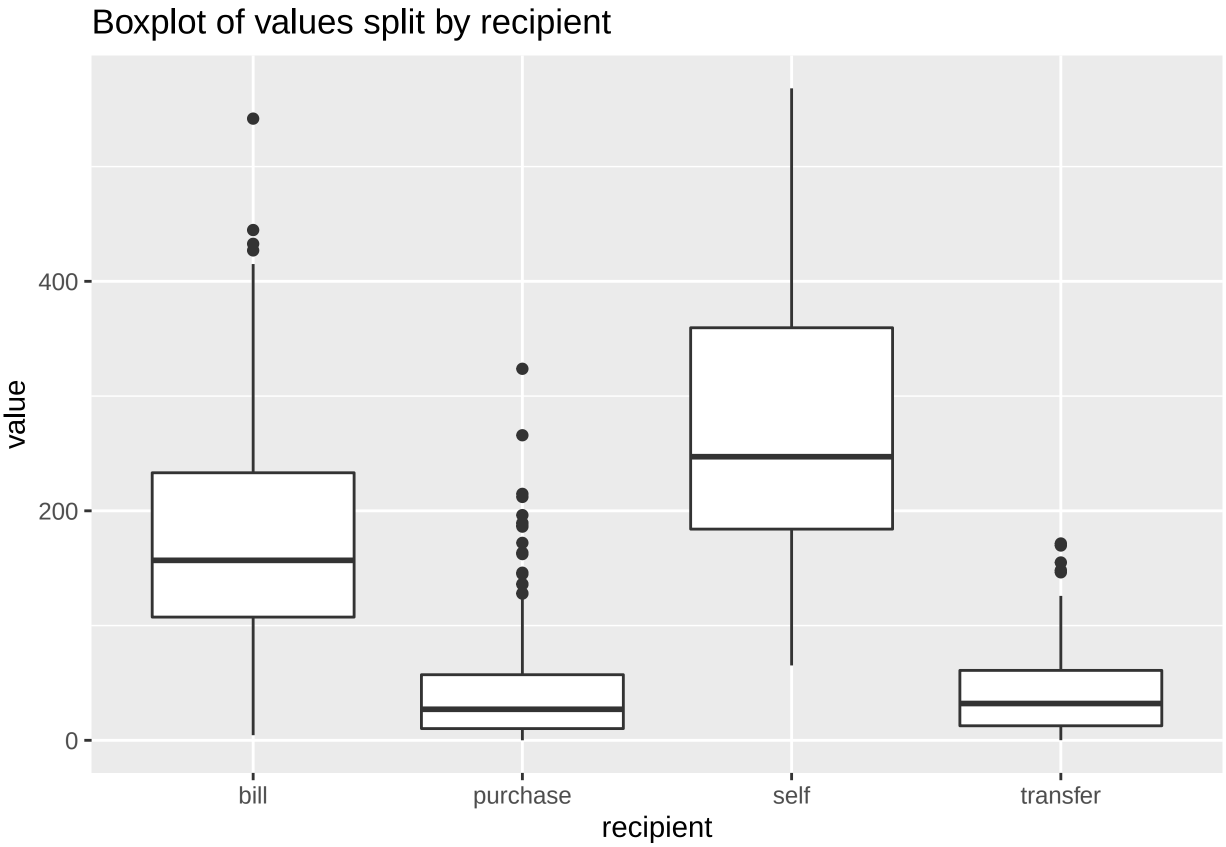
```
## [1] 96.24213
```

```
median(dataset$value)
```

```
## [1] 51.985
```

1.3: value is highest for type self of recipient with mean 278 and standard deviation 135. Bill value of mean for recipient is 178 and standard deviation is 102.8 that will round up to 103. While, the purchase and transfer are the lowest with mean and standard deviation values.

```
# Boxplot of value variable split by the recipient category
ggplot(data = dataset, aes(y = value, x = recipient)) + geom_boxplot()+labs(title = "Boxplot of values s
```



```
# mean and standard deviation of a value  calculated within subgroups created by a recipient variable:
summary(dataset$value)
```

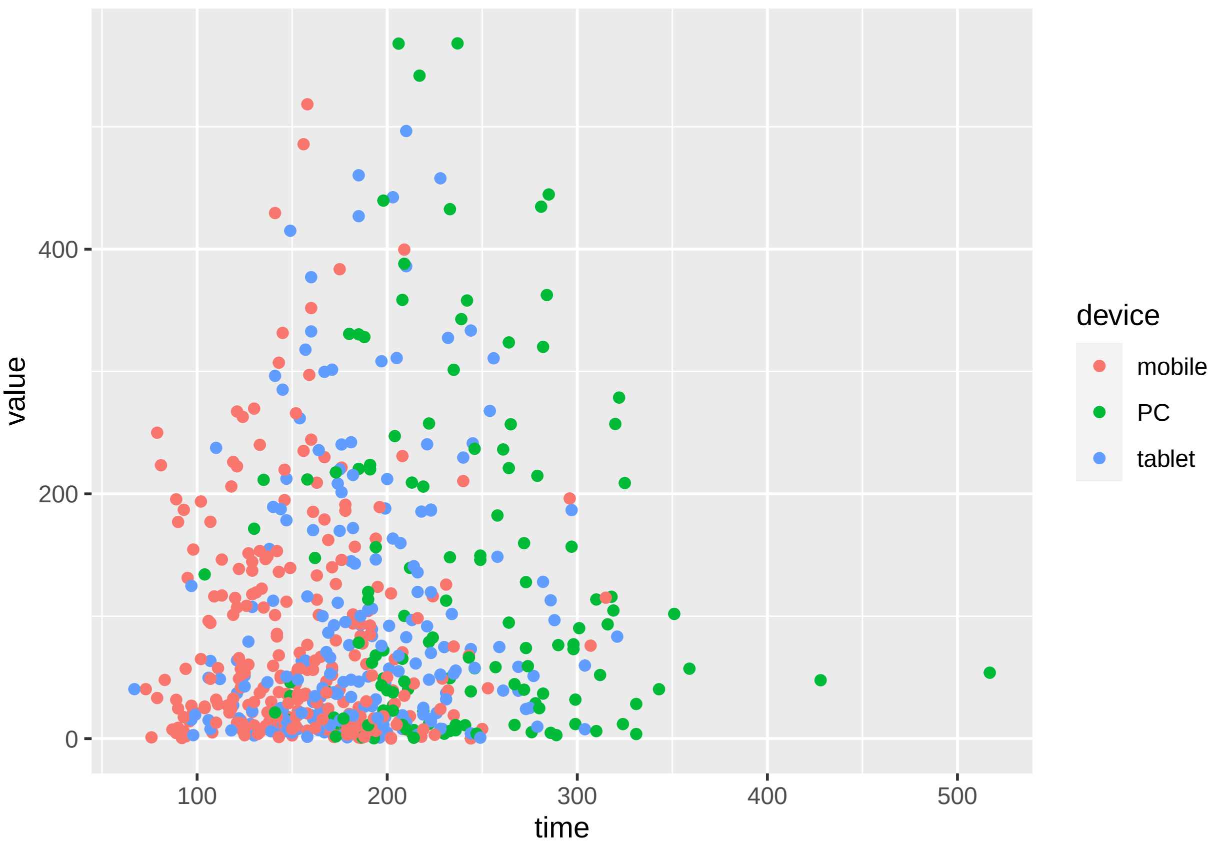
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.02   17.18   51.98   96.24  140.27  568.18
```

```
summarise(group_by(dataset, recipient), means = mean(value),  stddev = sd(value))
```

```
## # A tibble: 4 x 3
##   recipient means stddev
##   <fct>     <dbl>  <dbl>
## 1 bill      178.   103.
## 2 purchase   42.4   46.6
## 3 self      278.   135.
## 4 transfer   43.2   41.5
```

1.4: We can see a maximum number of transactions ranging between 0-50£ is highest and thereby increasing value of transaction pounds for money . We can see maximum of number of transactions has been done in time ranging between 30 - 250 seconds ranging from mobile, PC and tablet.

```
# Scatter plot of time against value with the points color coded according to the device
ggplot(data = dataset, aes(x = time, y = value, color = device)) + geom_point()
```



```
# Correlation of value and time
cor(dataset$value, dataset$time, method = 'pearson')
```

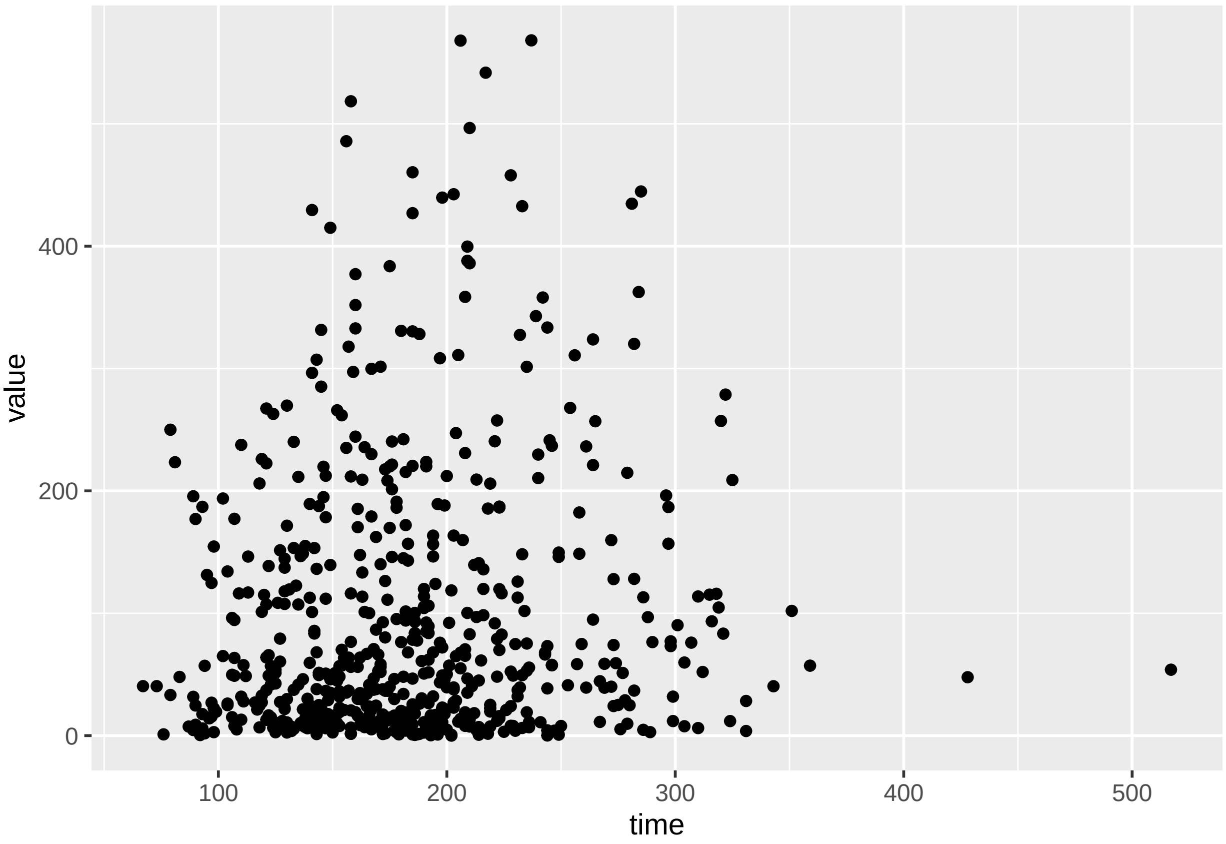
```
## [1] 0.1008659
```

```
cor(dataset$value, dataset$time, method = 'spearman')
```

```
## [1] 0.108048
```

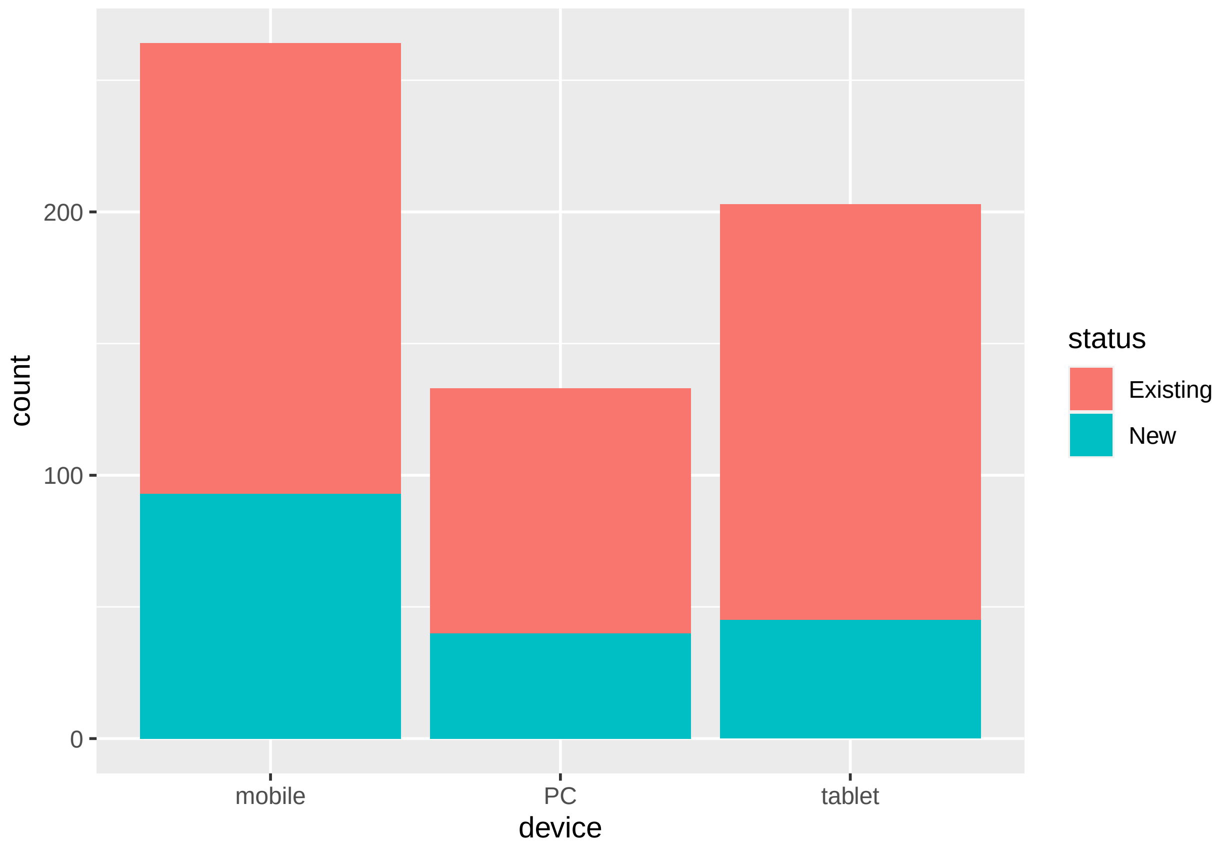
```
#cor(diamonds$price, diamonds$carat, method = 'spearman')
```

```
# Spearman higher than Pearson, so probably non-linear trend
ggplot(data = dataset, aes(x = time, y = value)) + geom_point()
```

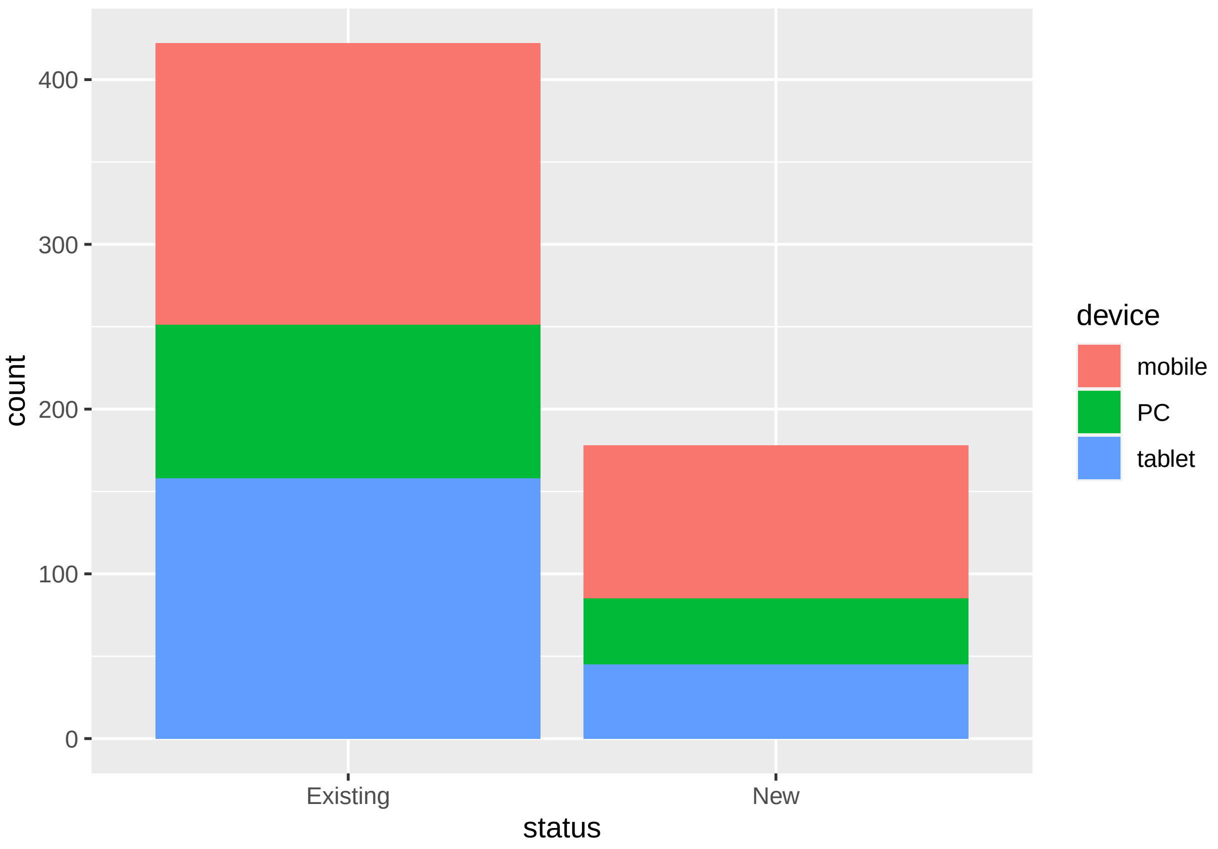


1.5: Stack bar plots with device against status shows that the ‘Existing’ category of status is higher than the ‘New’ in three of the devices. While stack bar plot for vice versa shows for ‘Existing’ category of status both mobile and tablet has same distributions while for ‘New’ category device mobile is maximum, tablet and PC is minimum spread

```
# Stack bar plots with device against status shows existing category of status is higher than the New i
ggplot(data = dataset, aes(x = device, fill = status)) + geom_bar()
```



```
# Stack barplots with status against device shows for existing category of status both mobile and table
ggplot(data = dataset, aes(x = status, fill = device)) + geom_bar()
```



SECTION 2: TABLES AND MEASURES:

2.1: One-way table shows that the Purchase is most common recipient type with 51.3% and self is least common recipient type with 8.7%

```
# Tables of recipient values
frequency = table(dataset$recipient) ; #frequency
frequency = sort(frequency, decr = T) ; #frequency
percentage = 100*round(prop.table(frequency),3) ; #percentage
total = cumsum(frequency) ; total
```

```
## purchase     bill transfer     self
##      308      455      548      600
```

```
cbind(Count = frequency, Percentage = 100*round(prop.table(frequency),3), Total = cumsum(frequency))
```

```
##          Count Percentage Total
## purchase   308       51.3   308
## bill       147       24.5   455
## transfer    93       15.5   548
## self        52        8.7   600
```

2.2: Two way table shows that the Purchase has much lower rate of Existing than average, however, self is higher than average rate of Existing. Whereas Bill and transfer are average and slightly less than self rate.

```
# Two way table of recipient(in rows) and status(in columns) normalised to show the fraction of status
RS2wayT = xtabs(~ recipient + status, data = dataset)
RS2wayT
```

```
##           status
## recipient  Existing New
##   bill          120  27
##   purchase      181 127
##   self           46   6
##   transfer       75  18
```

```
prop.table(RS2wayT, margin = 1) # Normalised
```

```
##           status
## recipient   Existing       New
##   bill     0.8163265 0.1836735
##   purchase 0.5876623 0.4123377
##   self     0.8846154 0.1153846
##   transfer 0.8064516 0.1935484
```

2.3: Table1 shows Mean value is higher in Existing status than New. Table2 shows PC has higher mean and followed with tablet and mobile.

```
# Table1 showing mean of model2 broken down by status
summarise(group_by(dataset, status), means = mean(model2))
```

```
## # A tibble: 2 x 2
##   status   means
##   <fct>    <dbl>
## 1 Existing  83.7
## 2 New       78.3
```

```
# Table2 showing the mean of model2 broken down by device
summarise(group_by(dataset, device), means = mean(model2))
```

```
## # A tibble: 3 x 2
##   device means
##   <fct>  <dbl>
## 1 mobile  79.7
## 2 PC      84.6
## 3 tablet  83.5
```

SECTION 3: SIGNIFICANCE TESTS:

3.1: Model1: Mean is 79.9 with 99% confidence mean is going to be in 79.5 to 80.4. So, confidence interval of 99% says model1 is going to be mean of 79.9 for sure. Model2: Mean is 82 with 99% confidence mean is going to be in 81.5 to 82.5. So, confidence interval of 99% says model2 is going to be mean of 82.5 for sure. Notes: Both significance tests mean with confidence interval 99% doesn't overlap and have a gap.

```r
# 99% confidence interval for the mean value of model1
t.test(dataset$model1, mu = mean(dataset$model1), conf.level = 0.99)
```

```
##
##  One Sample t-test
##
## data:  dataset$model1
## t = 0, df = 599, p-value = 1
## alternative hypothesis: true mean is not equal to 79.97367
## 99 percent confidence interval:
##  79.51758 80.42975
## sample estimates:
## mean of x
##  79.97367
```

```r
# 99% confidence interval for the mean value of model2
t.test(dataset$model2, mu = mean(dataset$model2), conf.level = 0.99)
```

```
##
##  One Sample t-test
##
## data:  dataset$model2
## t = 0, df = 599, p-value = 1
## alternative hypothesis: true mean is not equal to 82.07133
## 99 percent confidence interval:
##  81.55730 82.58537
## sample estimates:
## mean of x
##  82.07133
```

3.2: t.test for significance level of 0.05 when compared with model2 and model1 variable. To check if there is a difference we test with null hypothesis as 0, paired test and two sided with 95% confidence and significance level of 5%. Therefore p value is very very tiny and below significance level. So, null hypothesis has failed in this case. there is a mean of the differences 2.09

```r
# Paired t-test for model2 against model1
t.test(dataset$model2, dataset$model1, sig.level=0.05,
       mu=0, paired = T, alternative = "two.sided")
```

```
##
##  Paired t-test
##
## data:  dataset$model2 and dataset$model1
## t = 11.29, df = 599, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  1.732782 2.462552
## sample estimates:
## mean of the differences
##                2.097667
```

t.test with null hypothesis difference of 2 is resulting p value as 0.29 that is not lower than threshold that is our significance level 0.05. Also we have not meaningfully proven that difference is greater than 2

```r
# paired test for difference in sample means n>>2 so parametric test is applicable
t.test(dataset$model2, dataset$model1, sig.level=0.05,
       mu=2, paired = T, alternative = "greater")
```

```
##
##  Paired t-test
##
## data:  dataset$model2 and dataset$model1
## t = 0.52567, df = 599, p-value = 0.2997
## alternative hypothesis: true difference in means is greater than 2
## 95 percent confidence interval:
##  1.791591      Inf
## sample estimates:
## mean of the differences
##                 2.097667
```

3.3: for test 1 with status of recipients, a very small p value, strong evidence to reject the null hypothesis and accept the alternative that there is a significant difference in model2 means of existing and new recipients.For test 2 model 2 with device p value is again very very small and lower than threshold of 0.01, that's null hypothesis rejected and tells us there is a definite difference

```r
#Non parametric test for performance check of model 2 on New and existing recipients
# Creating subsets of new and existing and apply wilcox test
m1 = dataset$model2[dataset$status == 'New']
m2 = dataset$model2[dataset$status == 'Existing']

wilcox.test(m1,m2, paired = F,
            mu = 0, exact = FALSE, alternative = "two.sided", sig.level = 0.01)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  m1 and m2
## W = 13316, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
```

```r
# Performing second test with Kruskal-wallis test to achieve the same outcome for model2 with device
kruskal.test(model2 ~ device, data = dataset)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  model2 by device
## Kruskal-Wallis chi-squared = 113.93, df = 2, p-value < 2.2e-16
```

3.4: Checking for difference between different recipients of model2. Test of normality suggest our p value is lower than 0.95 so it is not normal and plot aov shows a little slight variation in plot but still cannot trust anova as the normality test is lower than 0.95

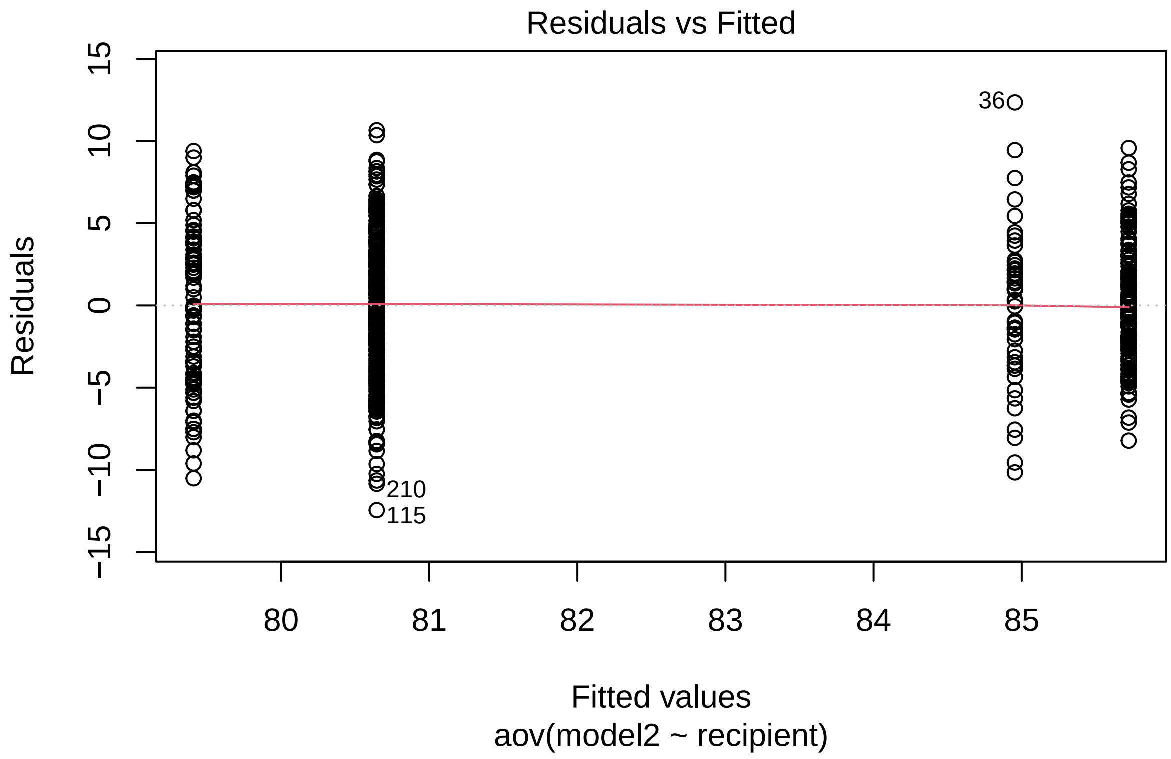
Anova test does suggest majorly a difference exists but not completely.

Tukey identifies differences between maximum levels, with transfer-bill the most different

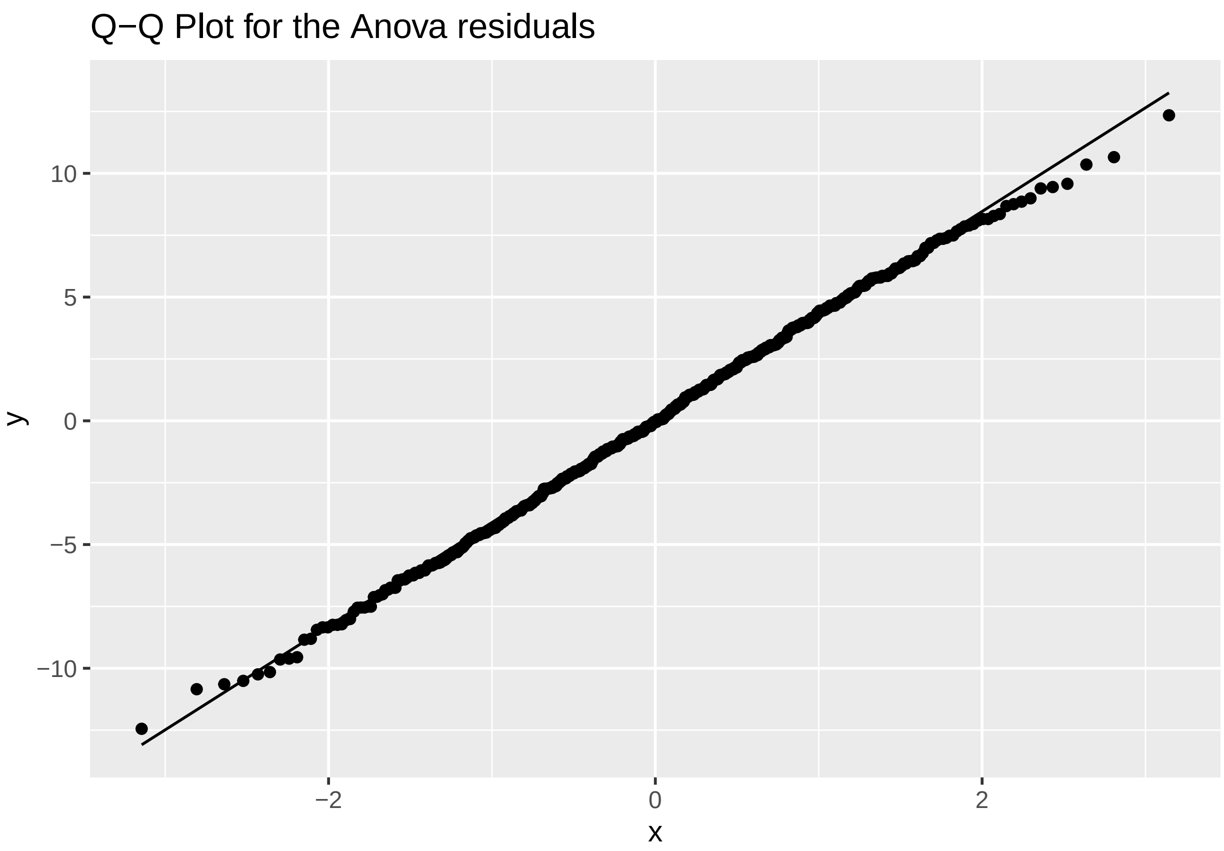
```
# Applying ANOVA, and checking residuals
anova <- aov(model2 ~ recipient, data = dataset)
summary(anova)
```

```
##                Df Sum Sq Mean Sq F value Pr(>F)    
## recipient     3   3676  1225.5   69.26 <2e-16 ***
## Residuals   596  10546    17.7                   
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
plot(anova, 1)
```



```
res = data.frame(residuals = anova$residuals)
ggplot(res, aes(sample = residuals)) + stat_qq() + stat_qq_line() +
  labs(title = "Q-Q Plot for the Anova residuals")
```



```
shapiro.test(anova$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  anova$residuals
## W = 0.9984, p-value = 0.8662
```

```
TukeyHSD(anova, conf.level = 0.95)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = model2 ~ recipient, data = dataset)
##
## $recipient
##                         diff       lwr         upr    p adj
## purchase-bill     -5.0766698 -6.163043 -3.99029680 0.000000
## self-bill         -0.7686028 -2.517134  0.97992873 0.669582
## transfer-bill     -6.3127716 -7.748634 -4.87690939 0.000000
## self-purchase      4.3080669  2.683336  5.93279804 0.000000
## transfer-purchase -1.2361018 -2.518322  0.04611844 0.063504
## transfer-self     -5.5441687 -7.420667 -3.66767063 0.000000
```

Shapiro test of normality is not good so we are applying non parametric test for difference in medians. p value is very small and yes there is a significant difference. Pairwise wilcox test shows the difference is between almost all recipients except self-bill and transfer-purchase.

```
kruskal.test(model2 ~ recipient, data = dataset)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  model2 by recipient
## Kruskal-Wallis chi-squared = 155.84, df = 3, p-value < 2.2e-16
```

```
# To see where significance difference is lying
pairwise.wilcox.test(dataset$model2, dataset$recipient, exact = F, p.adjust.method = 'BH')
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  dataset$model2 and dataset$recipient
##
##          bill    purchase self
## purchase < 2e-16 -        -
## self     0.365   3.8e-09  -
## transfer < 2e-16 0.043    5.7e-09
##
## P value adjustment method: BH
```

SECTION 4: EXPERIMENT DESIGN, SAMPLE SIZES AND RANDOM SAMPLING

4.1: Standard deviation of model 2 is 4.8

```
# Calculating standard deviation of model2
stddev = sd(dataset$model2)
stddev
```

```
## [1] 4.87267
```

4.2: The output is showing estimate of number of data points as 126 so, we have to gather to get a test with the respective levels. The output has been rounded up with ceiling function to avoid decimal values and also unnecessary output.

```
# Power t.test and rounding up the values for n size of sample
number = ceiling(power.t.test(power = 0.9,
             delta = 2,
             sd = stddev,
             sig.level = 0.05,
             type = "two.sample",
             alternative = 'two.sided')$n)
number
```

```
## [1] 126
```

Sample size of data points 287 would be needed to increase the power of the test to 0.99 with significance level of 0.01. This has been calculated using the same power.t.test as above whereas the power is as required 0.99 and significance level is 0.01

```
# Increasing the power of a test to 0.99 to see the sample size
number2 = ceiling(power.t.test(power = 0.99,
             delta = 2,
             sd = stddev,
             sig.level = 0.01,
             type = "two.sample")$n)
number2
```

```
## [1] 287
```

4.3: The dataframe of 126 random sample size as obtained in sample1 of 4.2 that has been produced by removing duplicates and slice sampling. And also the distribution table is plotted for all and sample size. With this we can see complete dataset without duplicates of size 600 observations are higher in distribution for bill, self and transfer than random sample dataframe of size 126. Whereas, purchase is higher in random sample than complete dataset.

```
# Ensuring the reproducibility
set.seed(123)
n=12
#Removing duplicates
dataset1 <- dataset[!duplicated(dataset), ]
#Setting sample size as above obtained in variable number
sample1 = slice_sample(dataset1, n = number)

dataGrouped = group_by(dataset1, recipient)

N = nrow(dataset1)

str(sample1)
```

```
## 'data.frame':    126 obs. of  8 variables:
##  $ id       : int  415 463 179 526 195 118 299 229 244 14 ...
##  $ status   : Factor w/ 2 levels "Existing","New": 1 2 2 1 1 1 1 2 1 1 ...
##  $ device   : Factor w/ 3 levels "mobile","PC",..: 1 1 3 2 1 2 3 3 3 2 ...
##  $ recipient: Factor w/ 4 levels "bill","purchase",..: 2 2 2 3 2 1 2 2 2 2 ...
##  $ value    : num  17.51 9.79 7.67 439.69 35.15 ...
##  $ time     : int  145 181 304 198 156 217 190 188 206 193 ...
##  $ model1   : num  78.4 78.4 71.2 86.1 77.2 83.4 81.1 77.6 73.3 79.6 ...
##  $ model2   : num  85.2 79.6 71.8 88.6 79.6 85.3 85.3 76.1 85.6 86.1 ...
```

```
t = rbind(complete = prop.table(table(dataset1$recipient)), sample = prop.table(table(sample1$recipient

t
```

```
##               bill  purchase       self  transfer
## complete 0.2450000 0.5133333 0.08666667 0.1550000
## sample   0.2142857 0.6111111 0.04761905 0.1269841
```

4.4: Stratified sample matches the complete dataset with purchase highly distributed as 0.51 and self as the least with 0.09

```
set.seed(123)
sample2 = slice_sample(dataGrouped, prop = number/N)

t = rbind(all = prop.table(table(dataset1$recipient)), Stratified = prop.table(table(dataset1$sample2$re
t
```

```
##      bill  purchase       self transfer
## all 0.245 0.5133333 0.08666667    0.155
```

```
round(t,2)
```

```
##     bill purchase self transfer
## all 0.24     0.51 0.09     0.16
```

Bibliography

Lecture and Lab notes - David Lonie, 2021-2022. Lecture 5, Lecture 6, Lecture 7, Lecture 8, Lecture 9, Lab 5, Lab 6, Lab7, Lab 8, Lab 9. Aberdeen: Robert Gordon University.