Project on UPI banks

Debi

26/04/2022

Load library, data cleaning etc

```r
# Load the libraries
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

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

```r
library(tidyr)
library(rstatix)
```

```
## Warning: package 'rstatix' was built under R version 4.1.3

##
## Attaching package: 'rstatix'

## The following object is masked from 'package:stats':
##
##     filter
```

```r
library(ggpubr)
```

```
## Warning: package 'ggpubr' was built under R version 4.1.3
```

```r
library(stringr)
```

```r
# Set working directory
setwd("C:/Users/System Administrator/Desktop/UPI PROJECT")

# Load the data
upi <- read.csv("./data/UPI apps transaction data in 2021.csv")
```

```r
# View the data
# View(upi)
upi %>% head()

# Change the variable name to easily work with
# UPI Banks -> upi_banks
# Volume Mn By Costumers -> cvol_mn
# Volume Cr By Costumers -> cval_cr
# Volume Mn -> vol_mn
# Volume Cr -> val_cr
# Month -> month
# Year -> year

# Change the column names
names(upi) <- c('upi_bank','cvol_mn','cval_cr','vol_mn','val_cr','month','year')

# Lets check variable data type
upi %>% glimpse()

# change the bank name as factor
upi$upi_bank <- as.factor(upi$upi_bank)
upi$month <- factor(upi$month, labels = month.abb, ordered = T)

# change month and year as date time format
upi$year <- year(upi$year)
upi$month <- month(upi$month)

# Lets check variable data type
upi %>% glimpse()

# How many year's data is there
unique(upi$year)
# As there are one year 2021, then we can remove the year column
upi <- upi %>% select(-'year')

# are there are any missing value
sum(is.na(upi))

# How many banks are available
length(levels(upi$upi_bank))

# dimension of the data
dim(upi)
# ----------------------- Data Cleaning End -----------------------------------
```

Data analysis

```r
# write.csv(upi, './data/upi_final_data.csv')
upi <- read.csv('./data/upi_final_data.csv')

# Let's check which bank have not 12 months data
banks <- upi %>%
```

```r
  group_by(upi_bank, month) %>%
  summarise(sum = n()) %>%
  summarise(count = sum(sum)) %>%
  filter(count == 12)
```

```
## `summarise()` has grouped output by 'upi_bank'. You can override using the
## `.groups` argument.
```

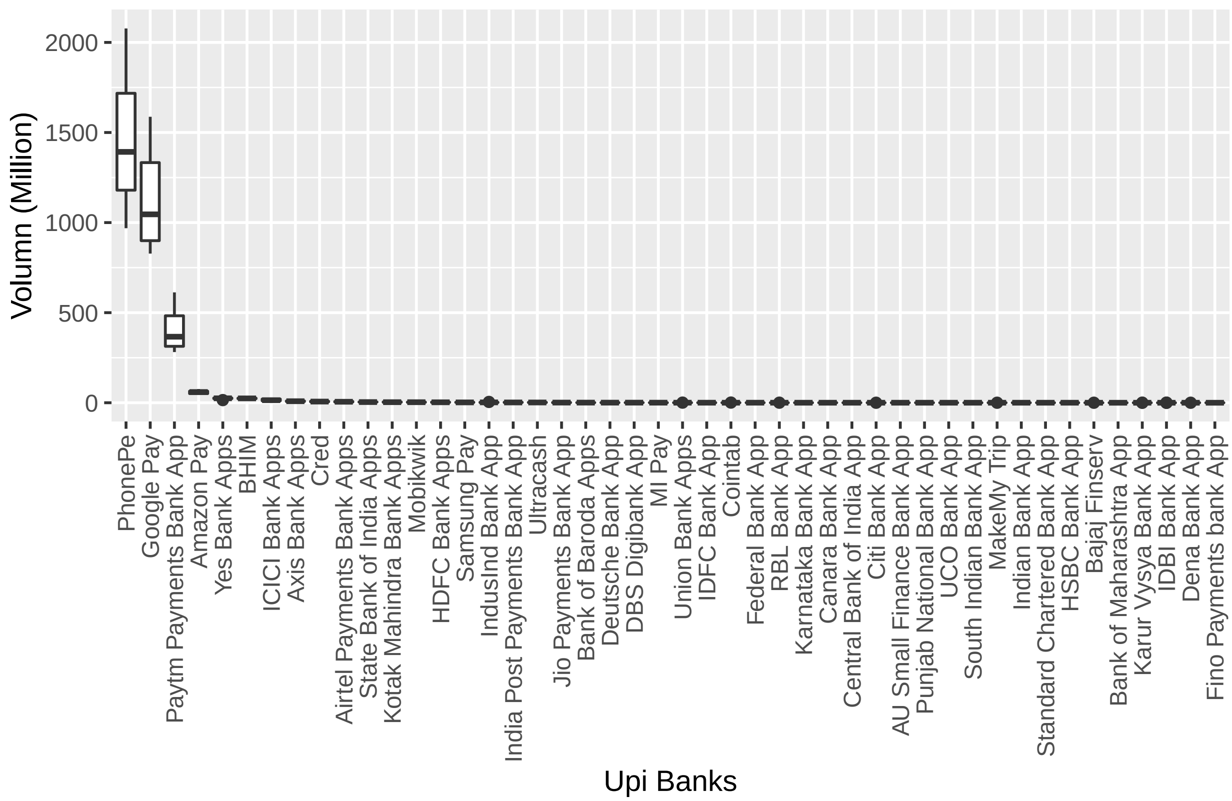
```r
upi <- upi %>%
  filter(upi_bank %in% banks$upi_bank)

# Top 4 bank have high volume
top_4_vol <- upi %>%
  group_by(upi_bank) %>%
  summarise(tot_vol = sum(cvol_mn)) %>%
  arrange(desc(tot_vol)) %>%
  top_n(4)
```

```
## Selecting by tot_vol
```

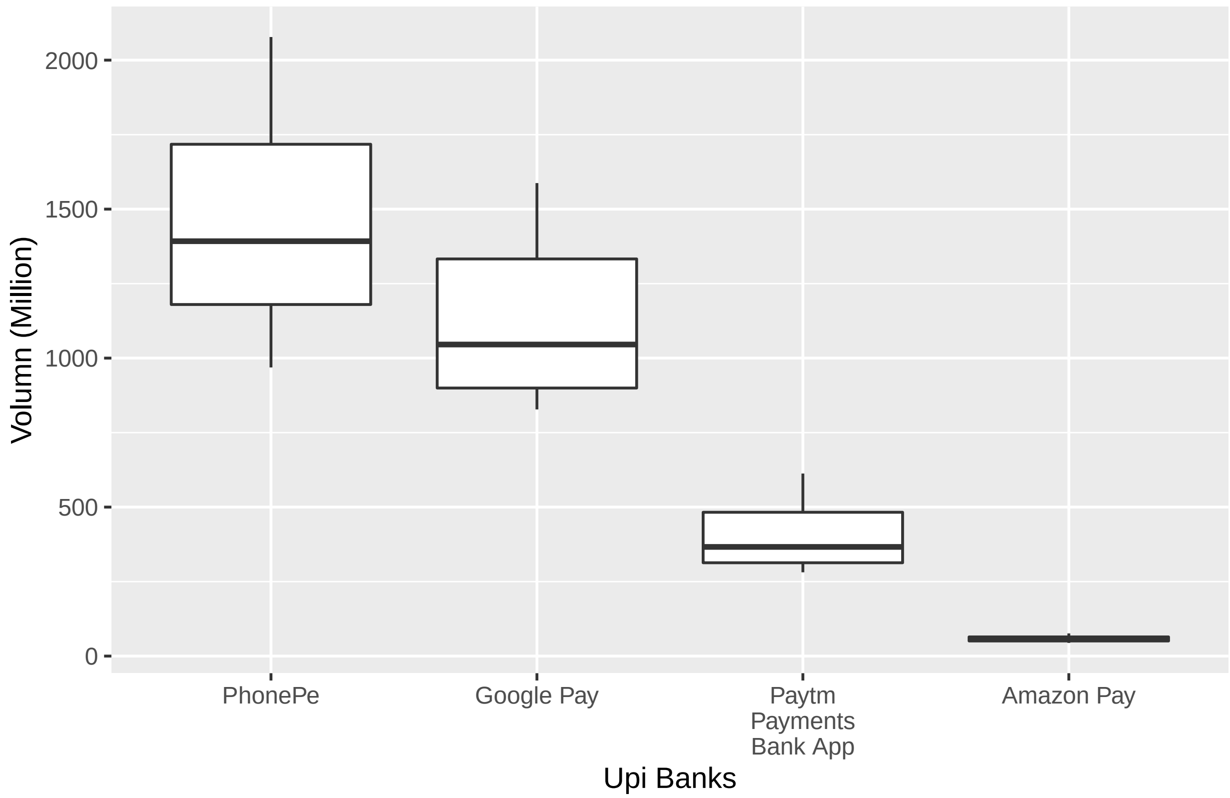
```r
# Top 4 bank have high value
top_4_value <- upi %>%
  group_by(upi_bank) %>%
  summarise(tot_value = sum(val_cr)) %>%
  arrange(desc(tot_value)) %>%
  top_n(4)
```

```
## Selecting by tot_value
```

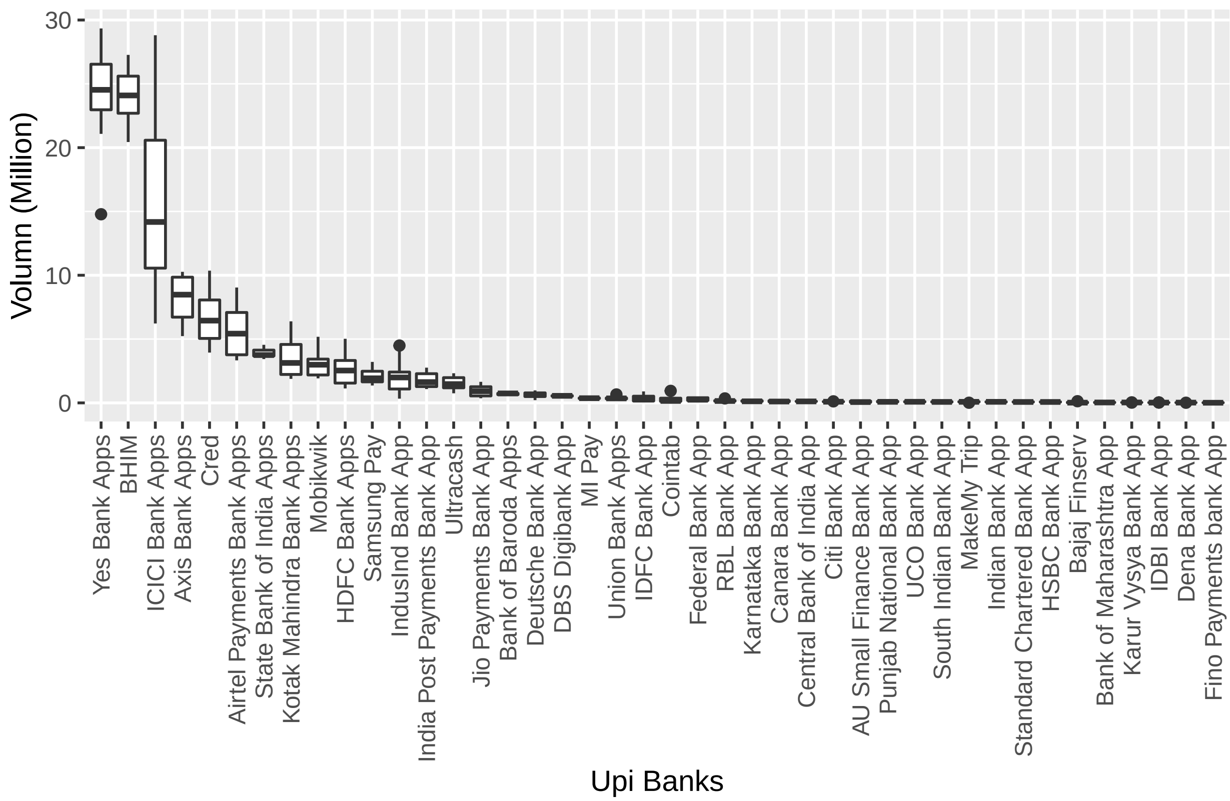
```r
# Plotting ---------------------------------------------------------------
# --------------------------- Volume ------------------------------------
# Boxplot of volumes
upi %>%
  group_by(upi_bank) %>%
  ggplot(aes(x = reorder(upi_bank, -cvol_mn), y = cvol_mn))+
  geom_boxplot() +
  labs(x = "Upi Banks", y = "Volumn (Million)",
       title = '') +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1),
        plot.title = element_text(hjust = 0.5))
```



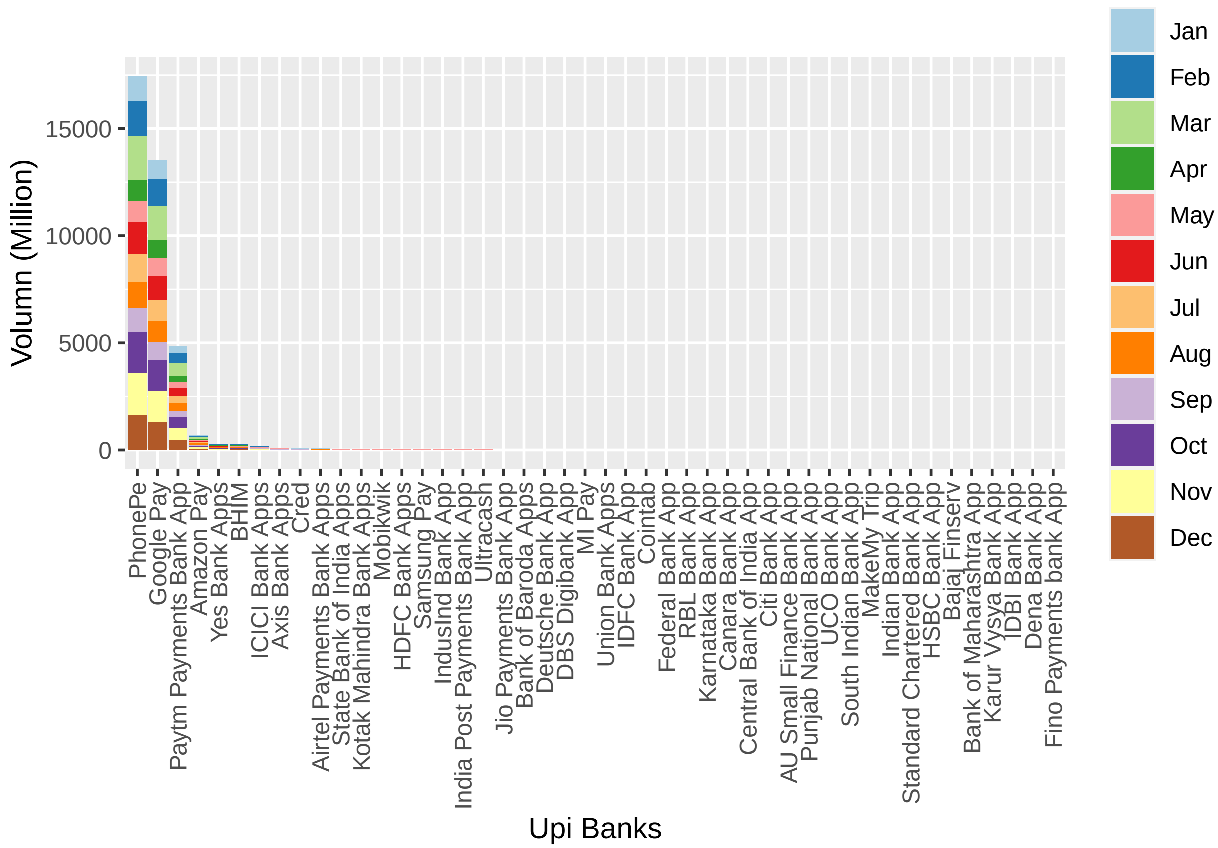

```
# Boxplot of top 4 banks having maximum volume
upi %>%
  group_by(upi_bank) %>%
  filter(upi_bank %in% top_4_vol$upi_bank) %>%
  ggplot(aes(x = reorder(upi_bank, -cvol_mn), y = cvol_mn))+
  geom_boxplot() +
  labs(x = "Upi Banks", y = "Volumn (Million)",
       title = '') +
  scale_x_discrete(labels = function(x) str_wrap(x, width = 10)) +
  theme(plot.title = element_text(hjust = 0.5))
```



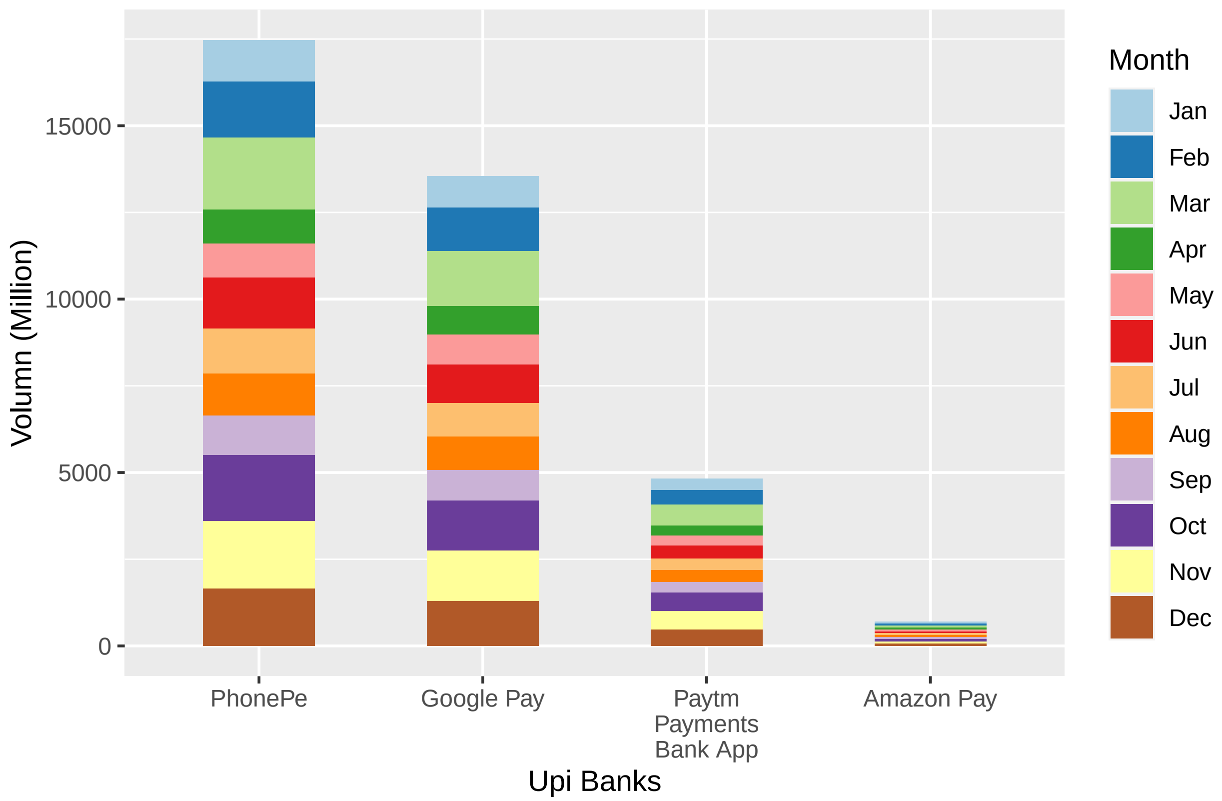

```
# Boxplot without top 4 banks
upi %>%
  group_by(upi_bank) %>%
  filter(!upi_bank %in% top_4_vol$upi_bank) %>%
  ggplot(aes(x = reorder(upi_bank, -cvol_mn), y = cvol_mn))+
  geom_boxplot() +
  labs(x = "Upi Banks", y = "Volumn (Million)",
       title = '') +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1),
        plot.title = element_text(hjust = 0.5))
```



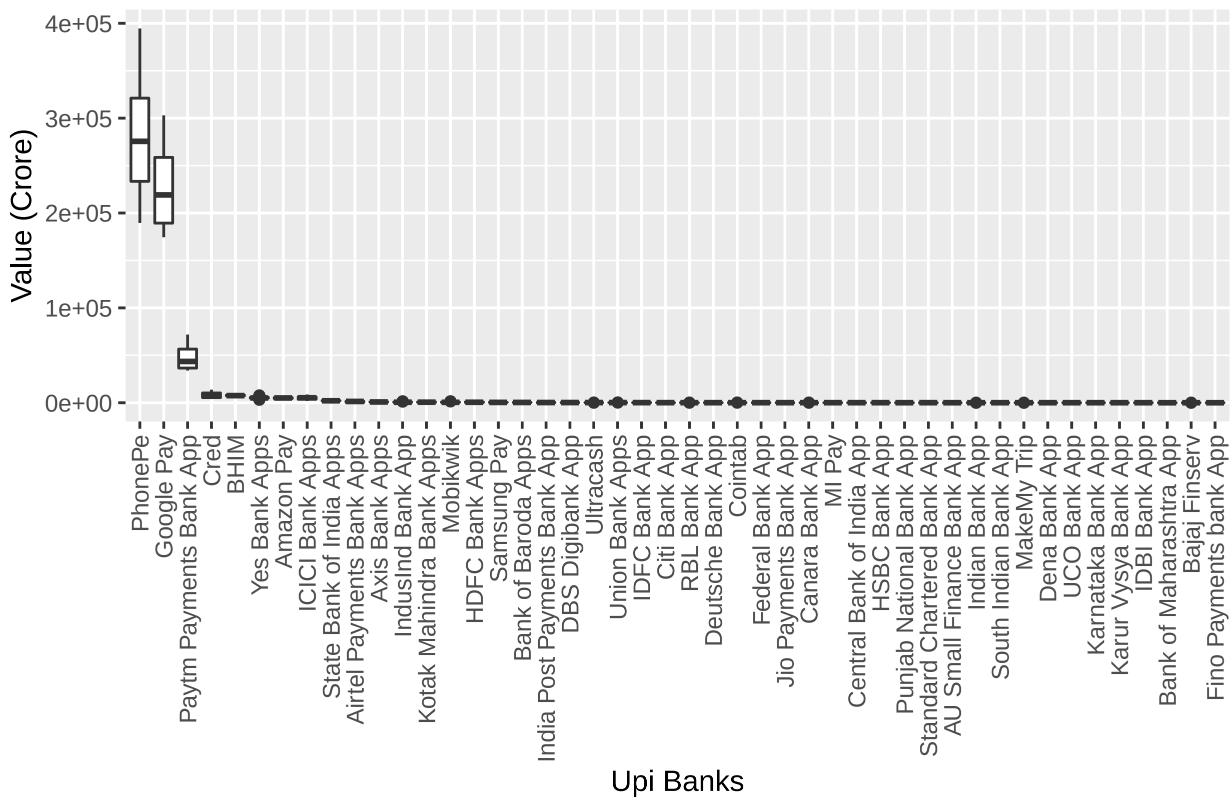

```
# Month wise comparison
upi %>%
  group_by(upi_bank) %>%
  ggplot(aes(x = reorder(upi_bank, -cvol_mn), y = cvol_mn,
             fill = factor(month, labels = month.abb))) +
  geom_bar(stat = 'identity', position = 'stack') +
  labs(x = "Upi Banks", y = "Volumn (Million)", fill = 'Month',
       title = '') +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1),
        plot.title = element_text(hjust = 0.5)) +
  scale_fill_brewer(palette = 'Paired')
```



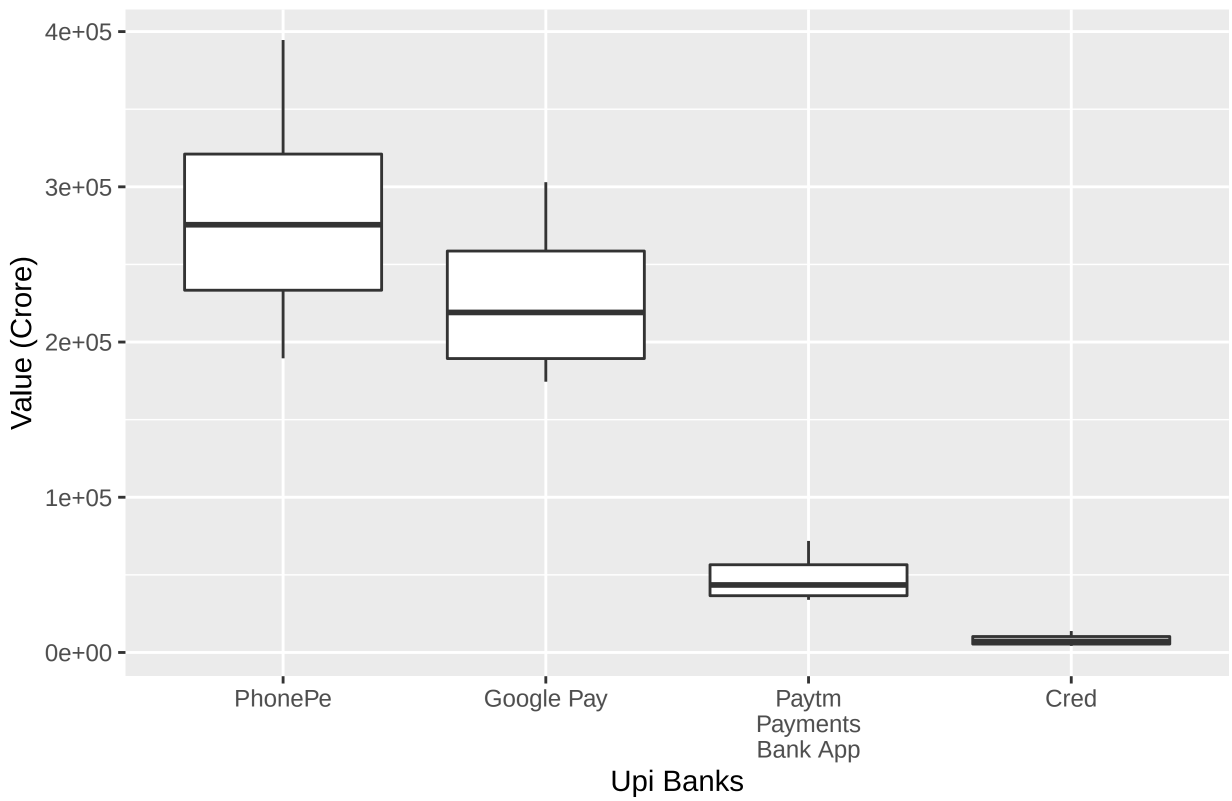

```r
# Month wise comparison of top 4 bank
upi %>%
  group_by(upi_bank) %>%
  filter(upi_bank %in% top_4_vol$upi_bank) %>%
  ggplot(aes(x = reorder(upi_bank, -cvol_mn), y = cvol_mn,
             fill = factor(month, labels = month.abb))) +
  geom_bar(stat = 'identity', position = 'stack', width = 0.5) +
  labs(x = "Upi Banks", y = "Volumn (Million)", fill = 'Month',
       title = '') +
  scale_x_discrete(labels = function(x) str_wrap(x, width = 10)) +
  theme(plot.title = element_text(hjust = 0.5)) +
  scale_fill_brewer(palette = 'Paired')
```



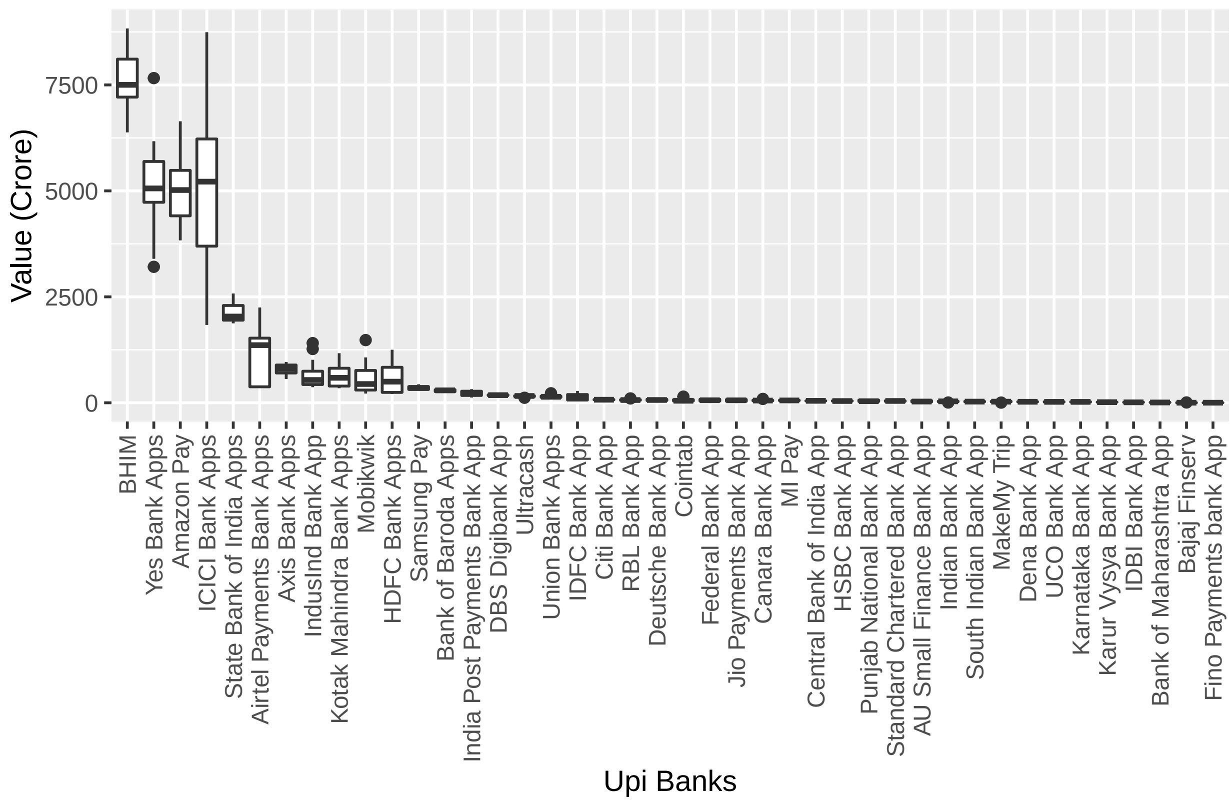

```
# ------------------------------- Value --------------------------------------
# Boxplot of Values
upi %>%
  group_by(upi_bank) %>%
  ggplot(aes(x = reorder(upi_bank, -cval_cr), y = cval_cr))+
  geom_boxplot() +
  labs(x = "Upi Banks", y = "Value (Crore)",
       title = '') +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1),
        plot.title = element_text(hjust = 0.5))
```



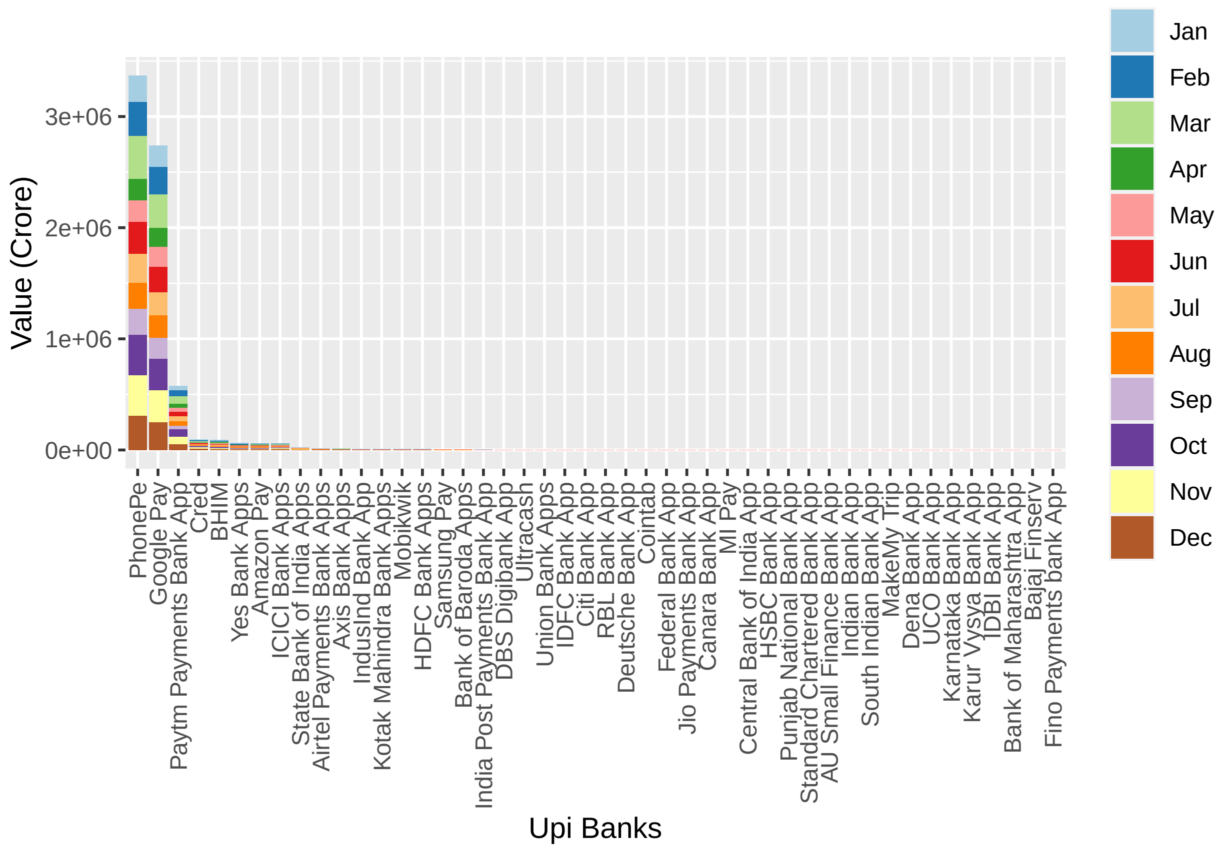

```r
# Boxplot of top 4 banks having maximum Value
upi %>%
  group_by(upi_bank) %>%
  filter(upi_bank %in% top_4_value$upi_bank) %>%
  ggplot(aes(x = reorder(upi_bank, -cval_cr), y = cval_cr))+
  geom_boxplot() +
  labs(x = "Upi Banks", y = "Value (Crore)",
       title = '') +
  scale_x_discrete(labels = function(x) str_wrap(x, width = 10)) +
  theme(plot.title = element_text(hjust = 0.5))
```



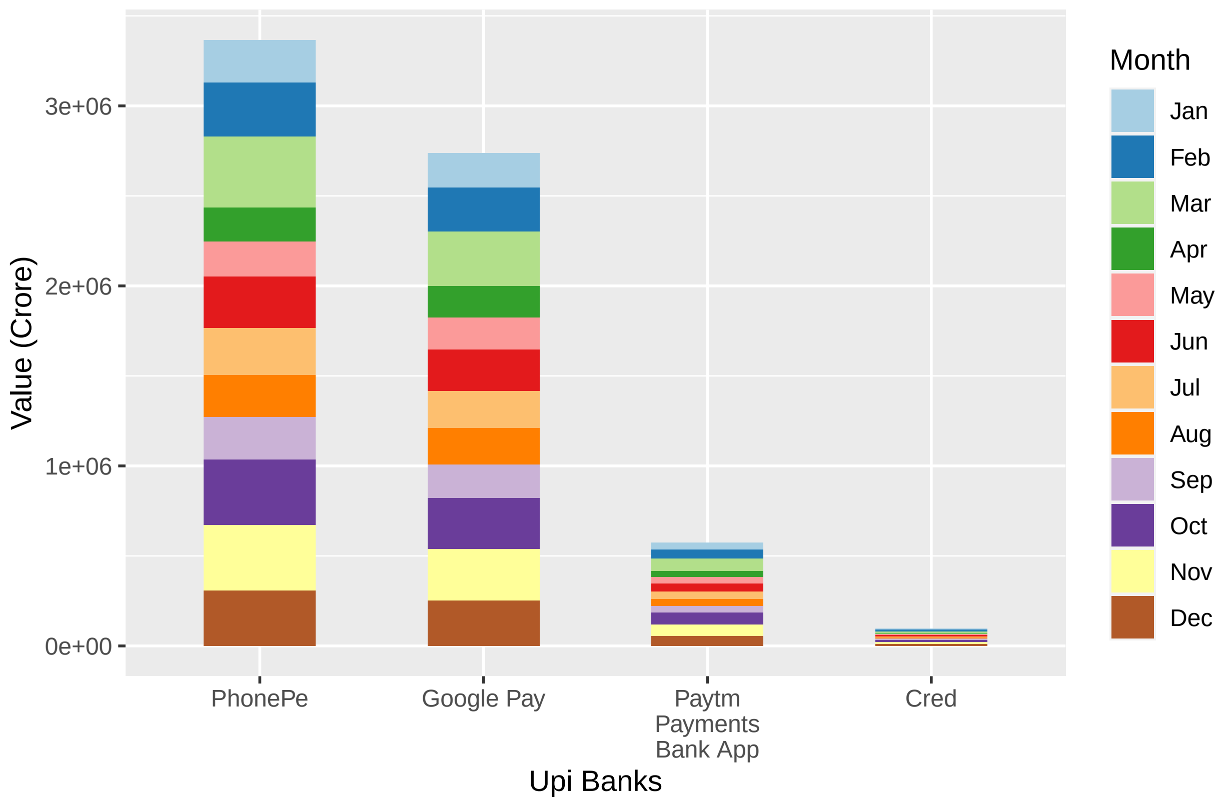

```r
# Boxplot without top 4 banks
upi %>%
  group_by(upi_bank) %>%
  filter(!upi_bank %in% top_4_value$upi_bank) %>%
  ggplot(aes(x = reorder(upi_bank, -cval_cr), y = cval_cr))+
  geom_boxplot() +
  labs(x = "Upi Banks", y = "Value (Crore)",
       title = '') +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1),
        plot.title = element_text(hjust = 0.5))
```




```
# Month wise comparison
upi %>%
  group_by(upi_bank) %>%
  ggplot(aes(x = reorder(upi_bank, -cval_cr), y = cval_cr,
             fill = factor(month, labels = month.abb))) +
  geom_bar(stat = 'identity', position = 'stack') +
  labs(x = "Upi Banks", y = "Value (Crore)", fill = 'Month',
       title = '') +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1),
        plot.title = element_text(hjust = 0.5)) +
  scale_fill_brewer(palette = 'Paired')
```



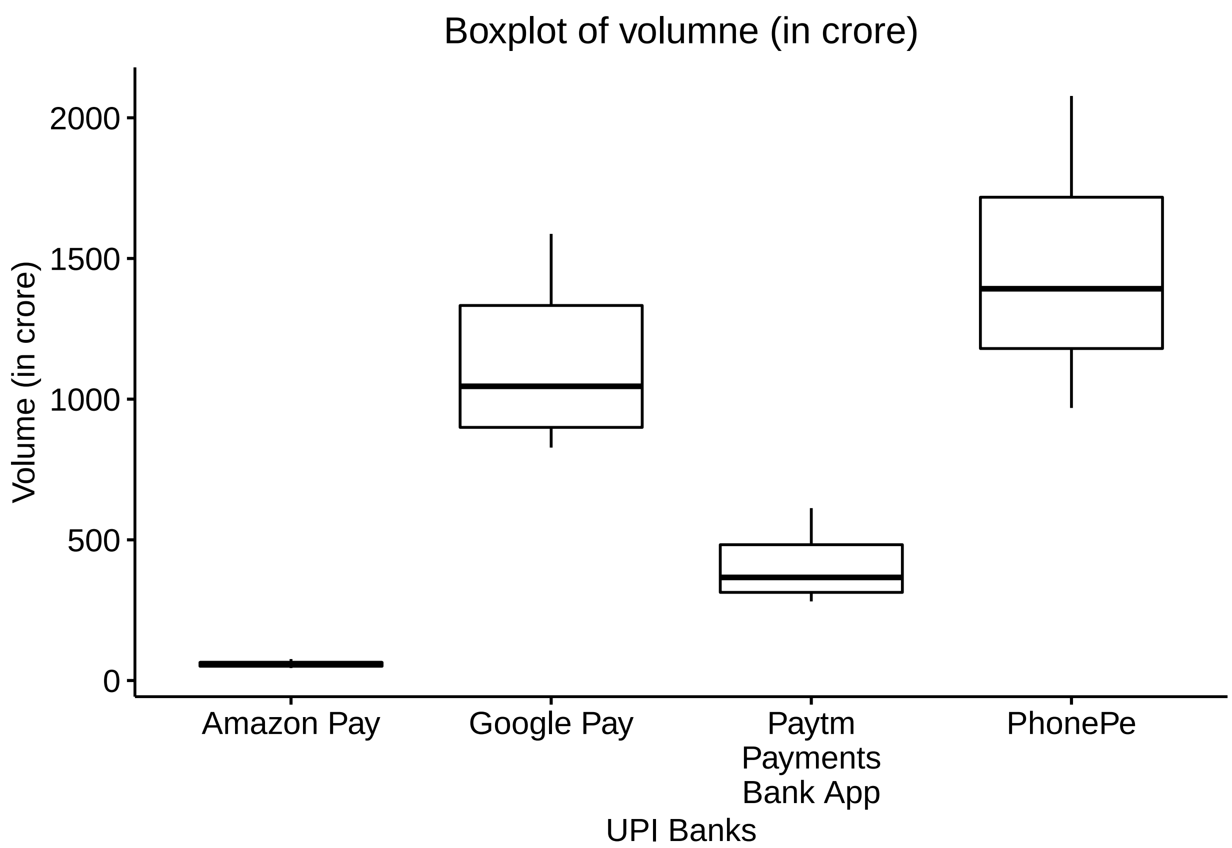

```
# Month wise comparison of top 4 bank
upi %>%
  group_by(upi_bank) %>%
  filter(upi_bank %in% top_4_value$upi_bank) %>%
  ggplot(aes(x = reorder(upi_bank, -cval_cr), y = cval_cr,
             fill = factor(month, labels = month.abb))) +
  geom_bar(stat = 'identity', position = 'stack', width = 0.5) +
  labs(x = "Upi Banks", y = "Value (Crore)", fill = 'Month',
       title = '') +
  scale_x_discrete(labels = function(x) str_wrap(x, width = 10)) +
  theme(plot.title = element_text(hjust = 0.5)) +
  scale_fill_brewer(palette = 'Paired')
```




```r
# ANOVA (volume)------------------------------------------------------------------

anova_vol <- upi %>%
  filter(upi_bank %in% top_4_vol$upi_bank) %>%
  select(upi_bank, month, cvol_mn)
# summary statistics
anova_vol %>%
  group_by(upi_bank) %>%
  get_summary_stats(cvol_mn, type = 'mean_sd')
```

```
## # A tibble: 4 x 5
##   upi_bank                variable     n   mean    sd
##   <chr>                   <chr>    <dbl>  <dbl> <dbl>
## 1 Amazon Pay              cvol_mn     12   58.7  10.2
## 2 Google Pay              cvol_mn     12 1128.   268.
## 3 Paytm Payments Bank App cvol_mn     12  402.   114.
## 4 PhonePe                 cvol_mn     12 1456.   384.
```

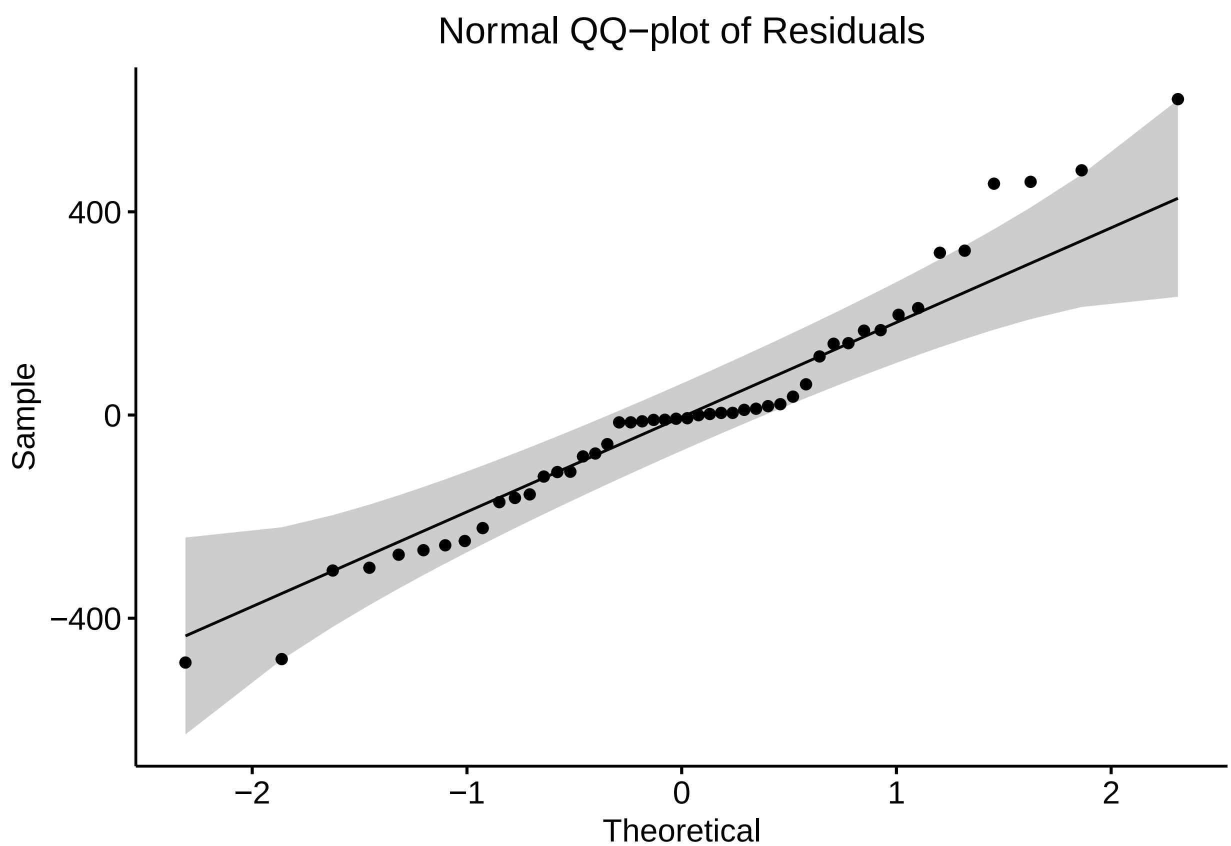
```r
# Visualization
ggboxplot(anova_vol, x = 'upi_bank', y = 'cvol_mn') +
  scale_x_discrete(labels = function(x) str_wrap(x,width = 10)) +
  labs(x = 'UPI Banks', y = 'Volume (in crore)', title = 'Boxplot of volumne (in crore)') +
  theme(plot.title = element_text(hjust = 0.5))
```




```r
# Identify outliers
anova_vol %>%
  group_by(upi_bank) %>%
  identify_outliers(cvol_mn)
```

```
## [1] upi_bank   month      cvol_mn    is.outlier is.extreme
## <0 rows> (or 0-length row.names)
```

```r
# There are no outlier in the data.

# Normality assumption (Model residual plot)
model <- lm(cvol_mn ~ upi_bank, data = anova_vol)
ggqqplot(residuals(model)) +
  labs(title = 'Normal QQ-plot of Residuals') +
  theme(plot.title = element_text(hjust = 0.5))
```




```
# In qqplot all the points fall approximately along the reference
# line. Also approximately all points are in the 2XSE region. So,
# the data statisfy the normality assumption.
shapiro_test(residuals(model))
```

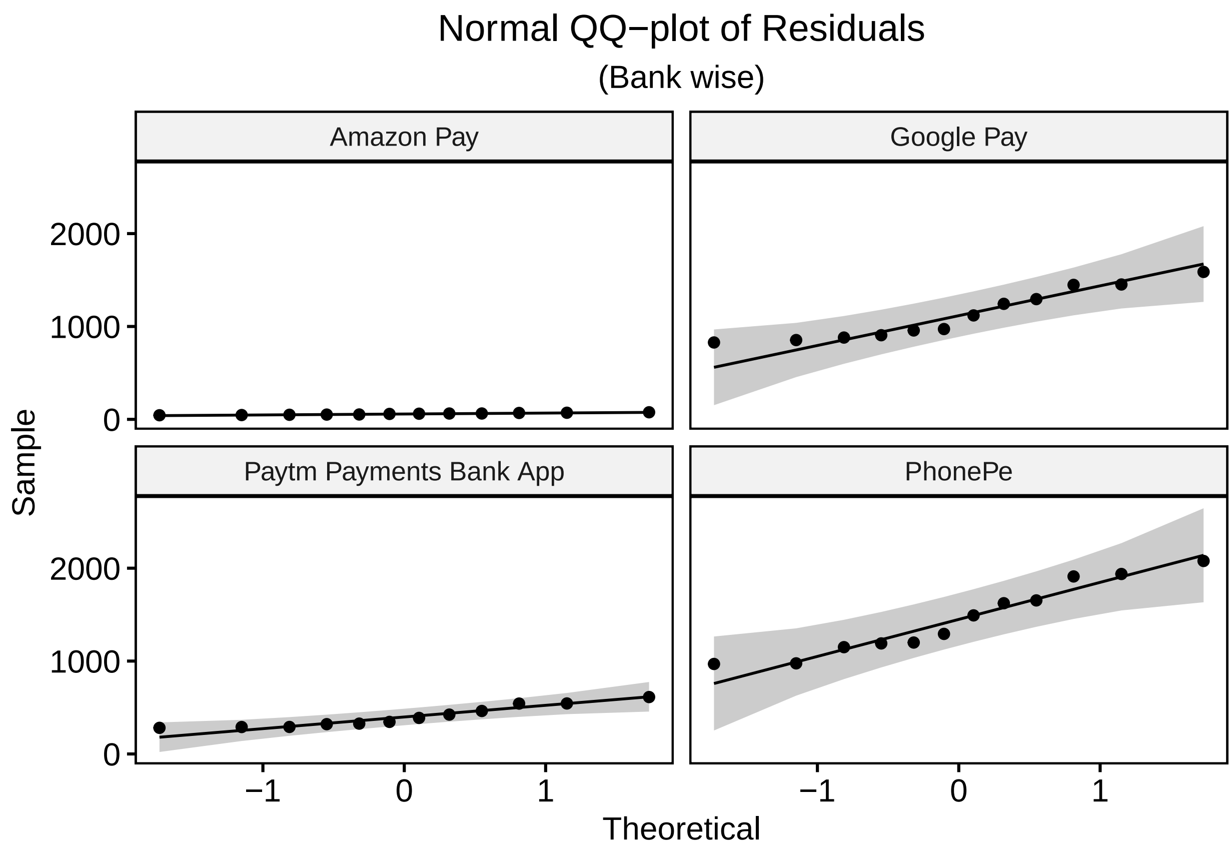
```
## # A tibble: 1 x 3
##   variable         statistic p.value
##   <chr>                <dbl>   <dbl>
## 1 residuals(model)     0.960   0.103
```

```
#---------------- Althogh sample size is not enough to do ------------------------
anova_vol %>%
  group_by(upi_bank) %>%
  shapiro_test(cvol_mn)
```

```
## # A tibble: 4 x 4
##   upi_bank                  variable statistic     p
##   <chr>                     <chr>        <dbl> <dbl>
## 1 Amazon Pay                cvol_mn      0.961 0.797
## 2 Google Pay                cvol_mn      0.896 0.139
## 3 Paytm Payments Bank App cvol_mn      0.891 0.121
## 4 PhonePe                   cvol_mn      0.926 0.339
```

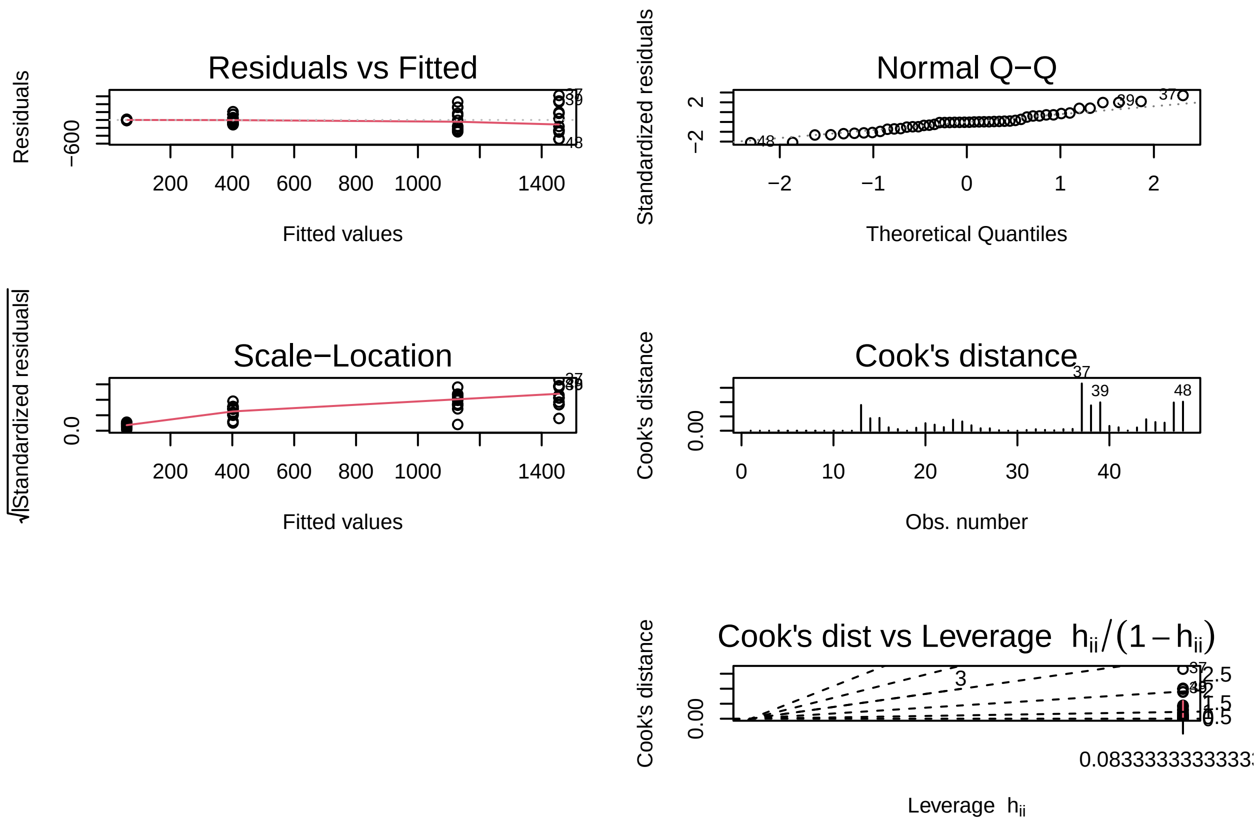
```
# All p-values are greater than 0.05, then we failed to reject the null hypothesis,
# we conclude that group-wise data is normally distributed.
ggqqplot(anova_vol, 'cvol_mn', facet.by = 'upi_bank') +
  labs(title = 'Normal QQ-plot of Residuals', subtitle = '(Bank wise)') +
  theme(plot.title = element_text(hjust = 0.5),
```

```r
        plot.subtitle = element_text(hjust = 0.5))
```




```r
# ---------------------------------------------------------------------------------

par(mfrow = c(3,2))
plot(model, 1:6)
```

```
## hat values (leverages) are all = 0.08333333
##  and there are no factor predictors; no plot no. 5
```




```r
# In the plot above, there is no evident relationships between residuals and
# fitted values (the mean of each groups), which is good. So, we can assume the
# homogeneity of variances.

summary(model)
```

```
##
## Call:
## lm(formula = cvol_mn ~ upi_bank, data = anova_vol)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -487.18 -129.88    -6.82   121.52   621.70
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                        58.68      69.53   0.844   0.4033
## upi_bankGoogle Pay               1069.81      98.33  10.880 4.65e-14 ***
## upi_bankPaytm Payments Bank App   343.60      98.33   3.494   0.0011 **
## upi_bankPhonePe                  1397.22      98.33  14.209  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 240.9 on 44 degrees of freedom
## Multiple R-squared:  0.8536, Adjusted R-squared:  0.8436
## F-statistic: 85.49 on 3 and 44 DF,  p-value: < 2.2e-16
```

```r
model
```

```
##
## Call:
## lm(formula = cvol_mn ~ upi_bank, data = anova_vol)
##
## Coefficients:
##                      (Intercept)               upi_bankGoogle Pay
##                            58.68                          1069.81
## upi_bankPaytm Payments Bank App                  upi_bankPhonePe
##                           343.60                          1397.22
```

```r
anova(model)
```

```
## Analysis of Variance Table
##
## Response: cvol_mn
##           Df   Sum Sq Mean Sq F value    Pr(>F)
## upi_bank   3 14878428 4959476  85.488 < 2.2e-16 ***
## Residuals 44  2552616   58014
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

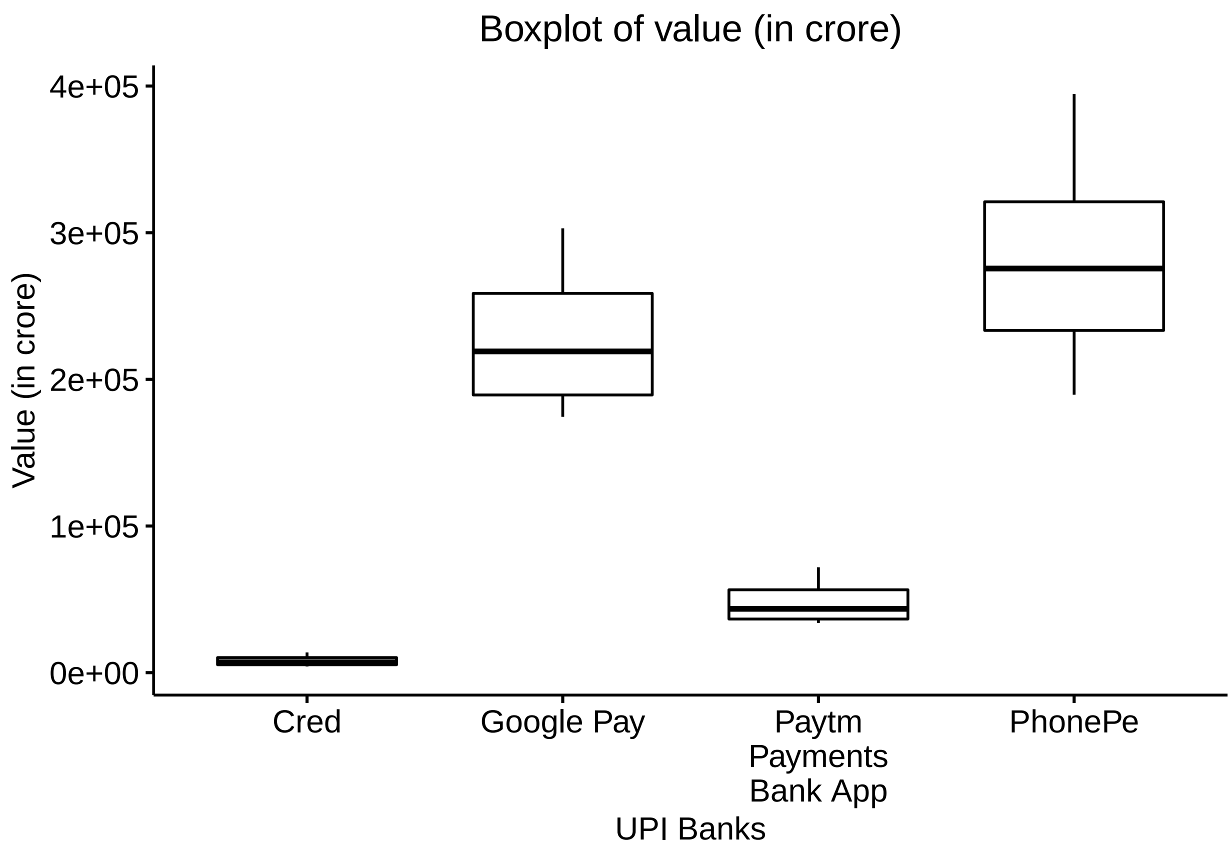
```r
# Post-hoc test
anova_vol %>% tukey_hsd(cvol_mn ~ upi_bank)
```

```
## # A tibble: 6 x 9
##   term     group1         group2 null.value estimate conf.low conf.high    p.adj
## * <chr>    <chr>          <chr>       <dbl>    <dbl>    <dbl>     <dbl>    <dbl>
## 1 upi_bank Amazon Pay     Googl~          0    1070.    807.       1332. 1.13e-12
## 2 upi_bank Amazon Pay     Paytm~          0     344.     81.1       606. 5.83e- 3
## 3 upi_bank Amazon Pay     Phone~          0    1397.   1135.       1660. 8.25e-13
## 4 upi_bank Google Pay     Paytm~          0    -726.   -989.       -464. 1.86e- 8
## 5 upi_bank Google Pay     Phone~          0     327.     64.9       590. 9.23e- 3
## 6 upi_bank Paytm Payment~ Phone~          0    1054.    791.       1316. 1.3 e-12
## # ... with 1 more variable: p.adj.signif <chr>
```

```r
# It can be seen form the output that all differences are significant.


# ANOVA (value)-----------------------------------------------------------------
anova_value <- upi %>%
  filter(upi_bank %in% top_4_value$upi_bank) %>%
  select(upi_bank, month, cval_cr)
# summary statistics
anova_value %>%
  group_by(upi_bank) %>%
  get_summary_stats(cval_cr , type = 'mean_sd')
```

```
## # A tibble: 4 x 5
##   upi_bank                variable     n    mean     sd
##   <chr>                   <chr>    <dbl>   <dbl>  <dbl>
## 1 Cred                    cval_cr     12   8084.  3374.
## 2 Google Pay              cval_cr     12 228125. 45376.
## 3 Paytm Payments Bank App cval_cr     12  47825. 13644.
## 4 PhonePe                 cval_cr     12 280477. 68612.
```

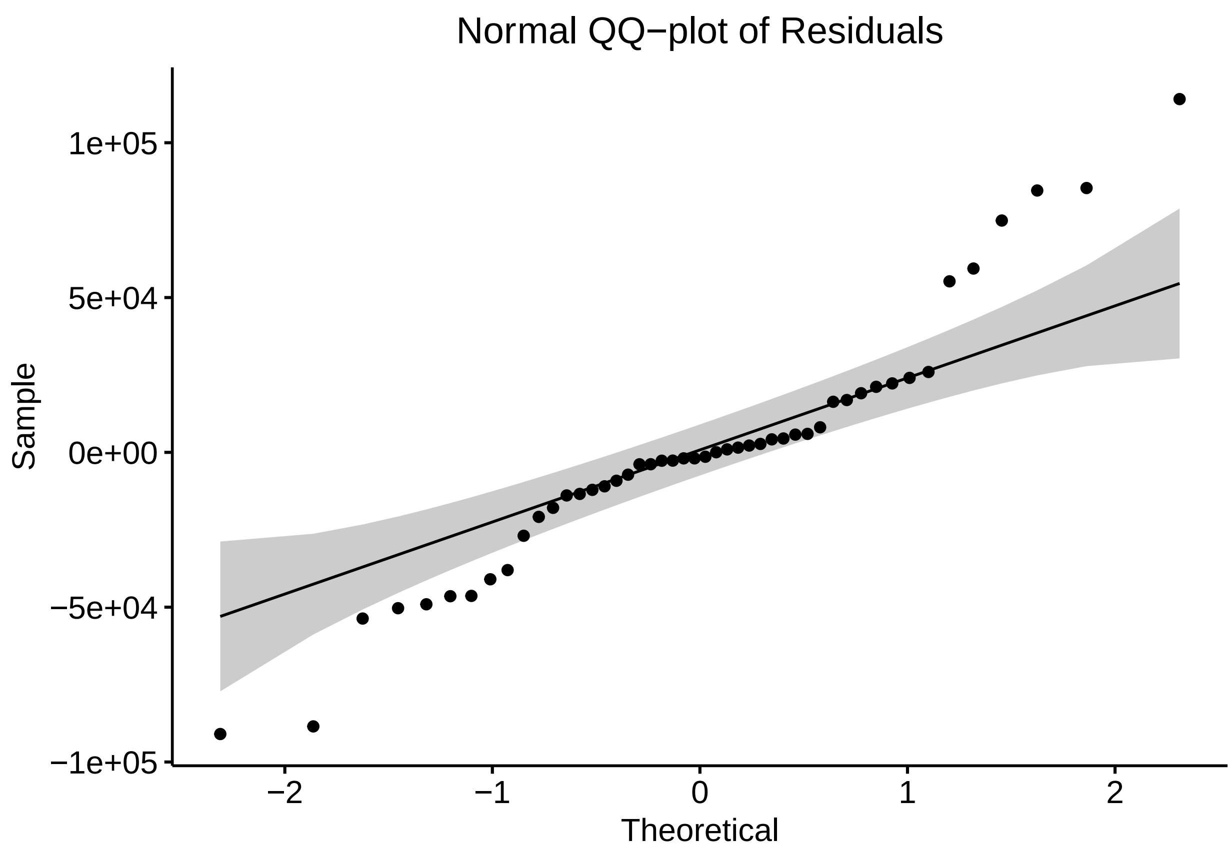
```r
# Visualization
ggboxplot(anova_value, x = 'upi_bank', y = 'cval_cr') +
  scale_x_discrete(labels = function(x) str_wrap(x,width = 10)) +
  labs(x = 'UPI Banks', y = 'Value (in crore)', title = 'Boxplot of value (in crore)') +
  theme(plot.title = element_text(hjust = 0.5))
```




```r
# Identify outliers
anova_value %>%
  group_by(upi_bank) %>%
  identify_outliers(cval_cr)
```

```
## [1] upi_bank   month      cval_cr    is.outlier is.extreme
## <0 rows> (or 0-length row.names)
```

```r
# There are no outlier in the data.

# Normality assumption (Model residual plot)
model <- lm(cval_cr  ~ upi_bank, data = anova_value)
ggqqplot(residuals(model)) +
  labs(title = 'Normal QQ-plot of Residuals') +
  theme(plot.title = element_text(hjust = 0.5))
```




```
# In qqplot all the points fall approximately along the reference
# line. Also approximately all points are in the 2XSE region. So,
# the data statisfy the normality assumption.
shapiro_test(residuals(model))
```

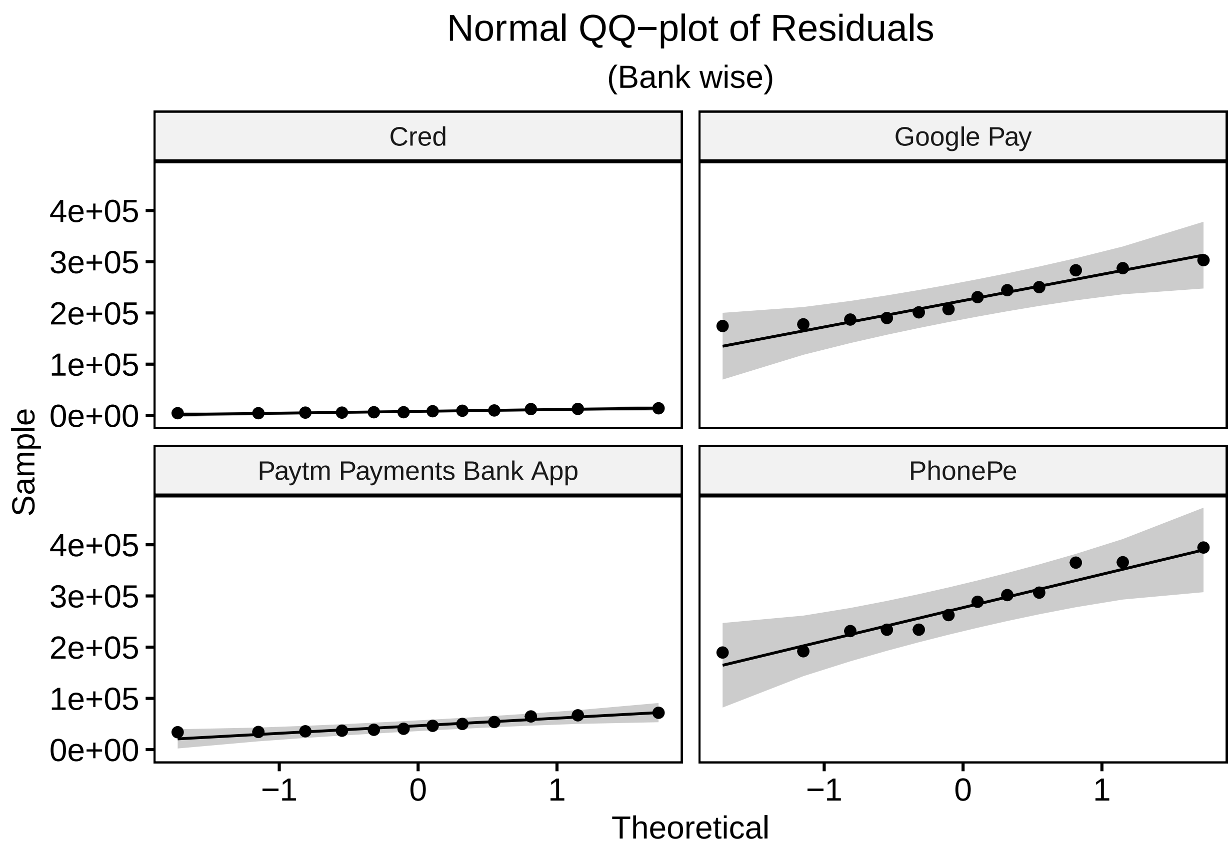
```
## # A tibble: 1 x 3
##   variable         statistic p.value
##   <chr>                <dbl>   <dbl>
## 1 residuals(model)     0.938  0.0142
```

```
#---------------- Althogh sample size is not enough to do -------------------------
anova_value %>%
  group_by(upi_bank) %>%
  shapiro_test(cval_cr)
```

```
## # A tibble: 4 x 4
##   upi_bank                variable statistic      p
##   <chr>                   <chr>        <dbl>  <dbl>
## 1 Cred                    cval_cr      0.901 0.165
## 2 Google Pay              cval_cr      0.912 0.225
## 3 Paytm Payments Bank App cval_cr      0.874 0.0745
## 4 PhonePe                 cval_cr      0.934 0.427
```

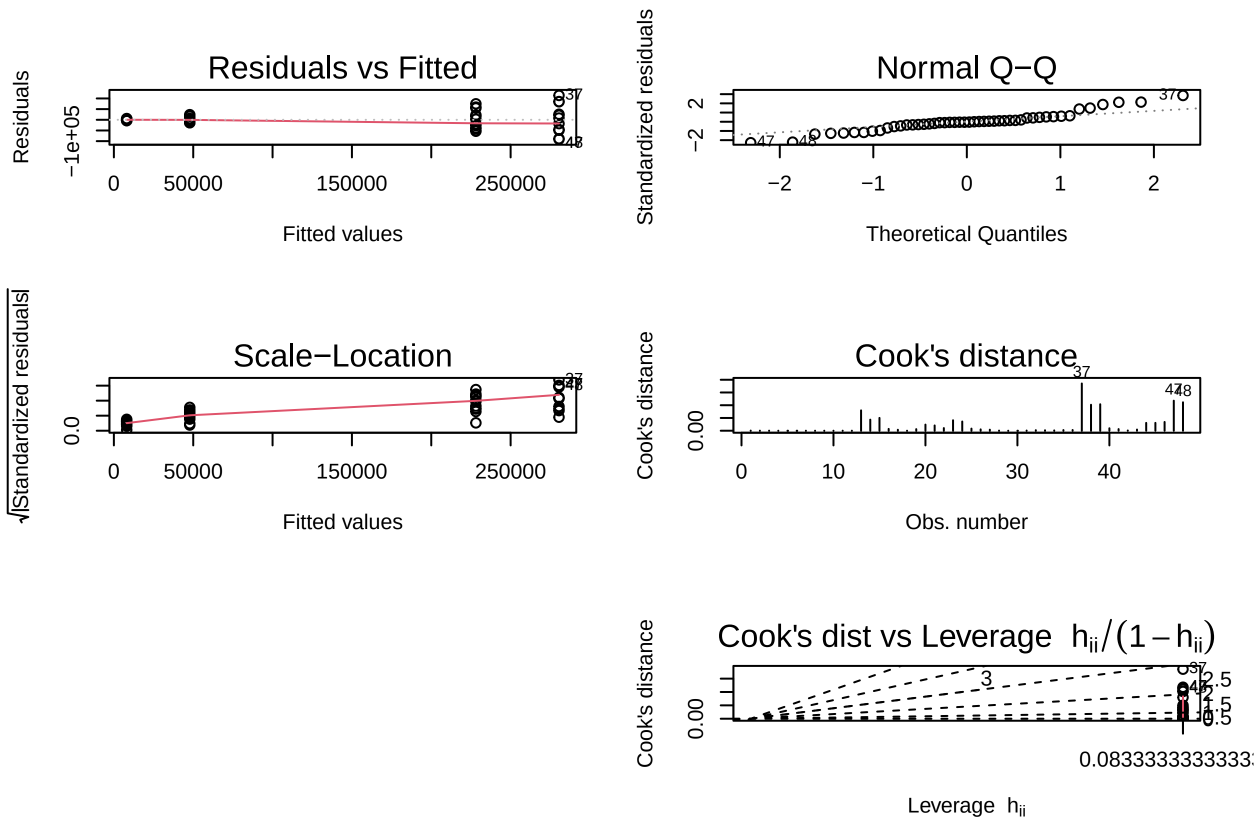
```
# All p-values are greater than 0.05, then we failed to reject the null hypothesis,
# we conclude that group-wise data is normally distributed.
ggqqplot(anova_value, 'cval_cr', facet.by = 'upi_bank') +
  labs(title = 'Normal QQ-plot of Residuals', subtitle = '(Bank wise)') +
  theme(plot.title = element_text(hjust = 0.5),
```

```
      plot.subtitle = element_text(hjust = 0.5))
```




```
# ----------------------------------------------------------------------------

par(mfrow = c(3,2))
plot(model, 1:6)
```

```
## hat values (leverages) are all = 0.08333333
##  and there are no factor predictors; no plot no. 5
```




```r
# In the plot above, there is no evident relationships between residuals and
# fitted values (the mean of each groups), which is good. So, we can assume the
# homogeneity of variances.

summary(model)
```

```
##
## Call:
## lm(formula = cval_cr ~ upi_bank, data = anova_value)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -90959 -14914  -1676  16472 114088
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                         8084      12045   0.671   0.5056
## upi_bankGoogle Pay                220040      17035  12.917   <2e-16 ***
## upi_bankPaytm Payments Bank App    39741      17035   2.333   0.0243 *
## upi_bankPhonePe                   272393      17035  15.991   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 41730 on 44 degrees of freedom
## Multiple R-squared:  0.8932, Adjusted R-squared:  0.8859
## F-statistic: 122.7 on 3 and 44 DF,  p-value: < 2.2e-16
```

```r
model
```

```
##
## Call:
## lm(formula = cval_cr ~ upi_bank, data = anova_value)
##
## Coefficients:
##                     (Intercept)              upi_bankGoogle Pay
##                            8084                          220040
## upi_bankPaytm Payments Bank App                upi_bankPhonePe
##                           39741                          272393
```

```r
anova(model)
```

```
## Analysis of Variance Table
##
## Response: cval_cr
##           Df     Sum Sq    Mean Sq F value    Pr(>F)
## upi_bank   3 6.4071e+11 2.1357e+11  122.67 < 2.2e-16 ***
## Residuals 44 7.6606e+10 1.7411e+09
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Post-hoc test
anova_value %>% tukey_hsd(cval_cr  ~ upi_bank)
```

```
## # A tibble: 6 x 9
##   term     group1         group2 null.value estimate conf.low conf.high    p.adj
## * <chr>    <chr>          <chr>       <dbl>    <dbl>    <dbl>     <dbl>    <dbl>
## 1 upi_bank Cred           Googl~          0  220040.  174558.   265523. 8.29e-13
## 2 upi_bank Cred           Paytm~          0   39741.   -5742.    85223. 1.06e- 1
## 3 upi_bank Cred           Phone~          0  272393.  226910.   317875. 8.25e-13
## 4 upi_bank Google Pay     Paytm~          0 -180300. -225782.  -134817. 1.52e-12
## 5 upi_bank Google Pay     Phone~          0   52353.    6870.    97835. 1.83e- 2
## 6 upi_bank Paytm Payment~ Phone~          0  232652.  187170.   278135. 8.27e-13
## # ... with 1 more variable: p.adj.signif <chr>
```

```r
# It can be seen form the output that all differences except between
# cred and paytm bank are significant.
```