Scatterplots

```r
library(ggplot2)
library(dplyr)
```

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

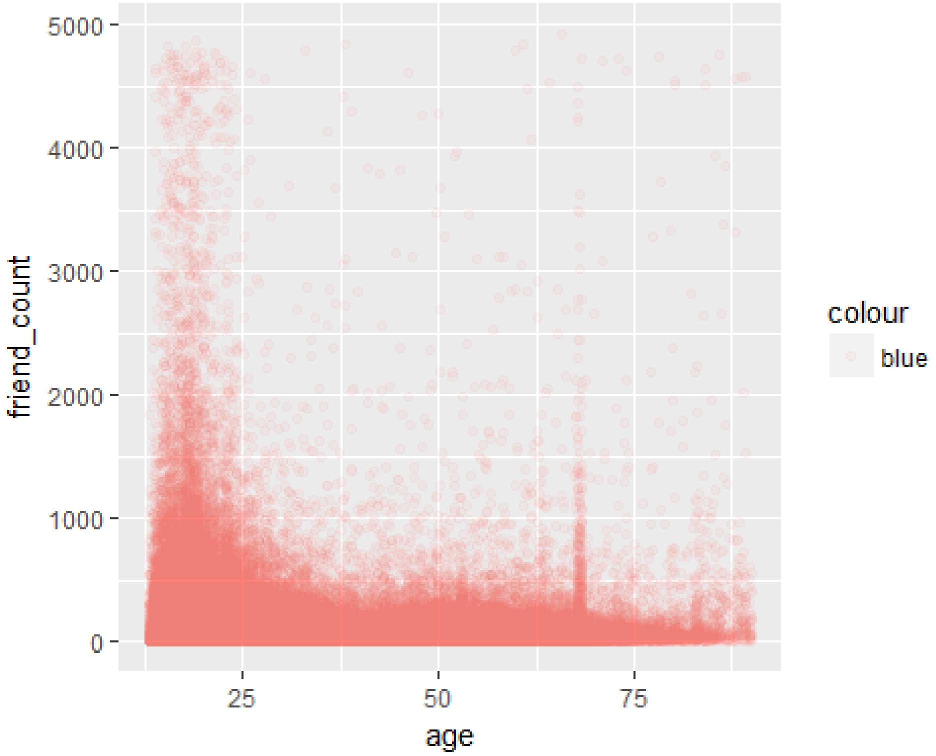
```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

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

```r
summary(pf$age)
```

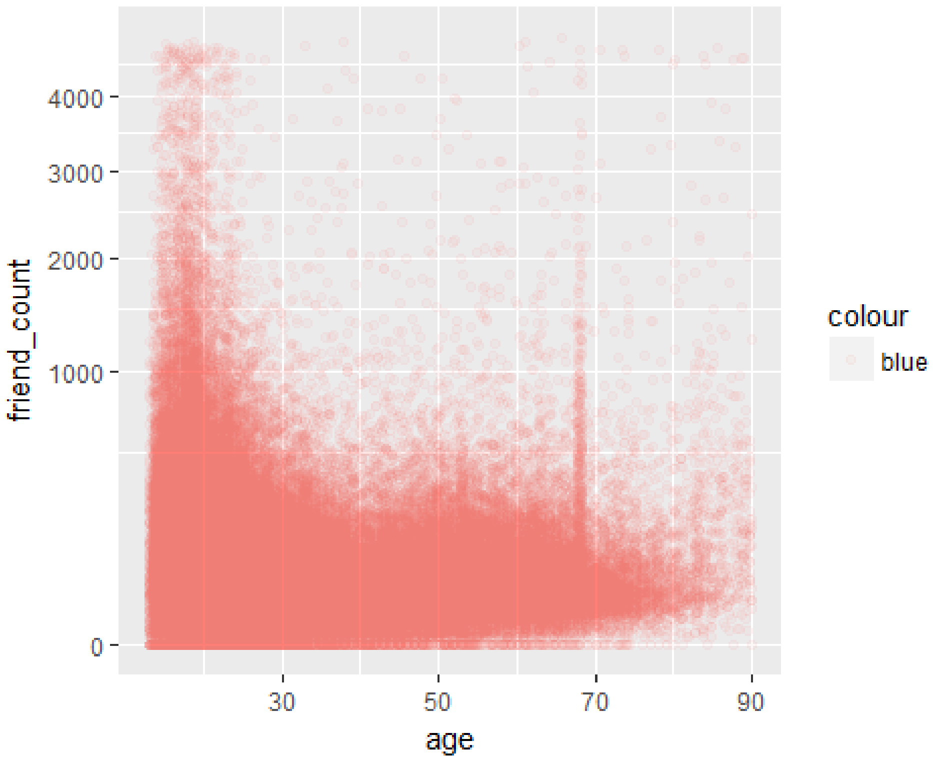
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   13.00   20.00   28.00   37.28   50.00  113.00
```

```
#The age variable becomes discrete so we use alpha 1/20 so we can have depth
and understand the numbers
ggplot(aes(x=age, y=friend_count, color='blue'), data=pf) +
  geom_jitter(alpha = 1/20) +
  xlim(13,90)

## Warning: Removed 5186 rows containing missing values (geom_point).
```

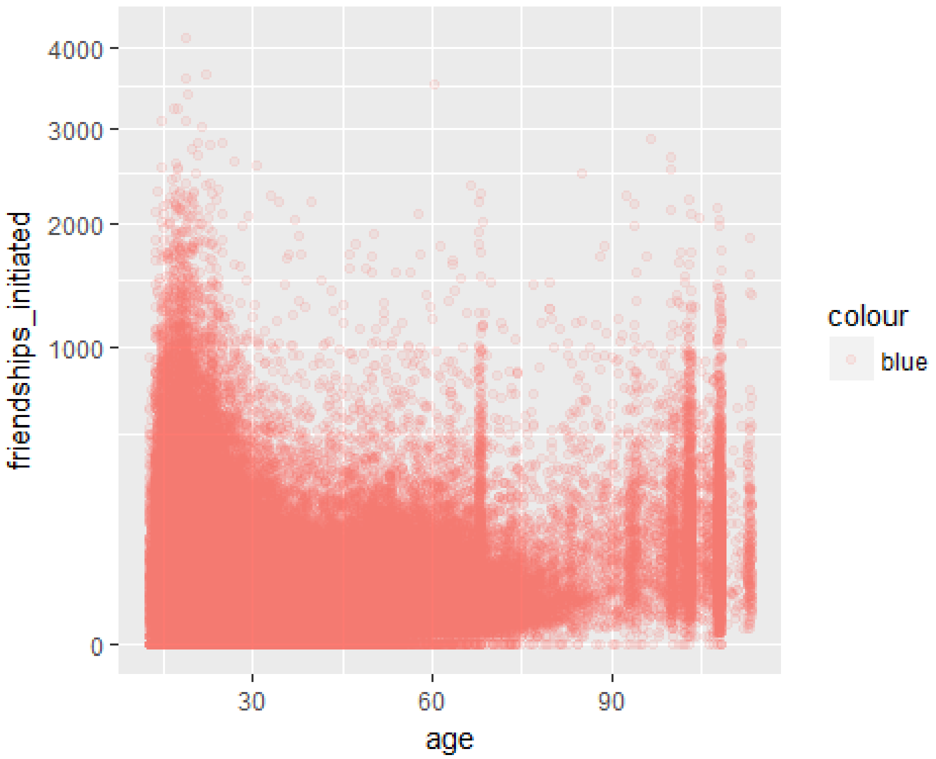


```
#Note we need to make sure the minimum height of the jitter to be 0 because w
e're using sqrt
ggplot(aes(x=age, y=friend_count, color='blue'), data=pf) +
  geom_jitter(alpha = 1/20, position=position_jitter(h = 0)) +
  xlim(13,90)+
  coord_trans(y="sqrt")
```

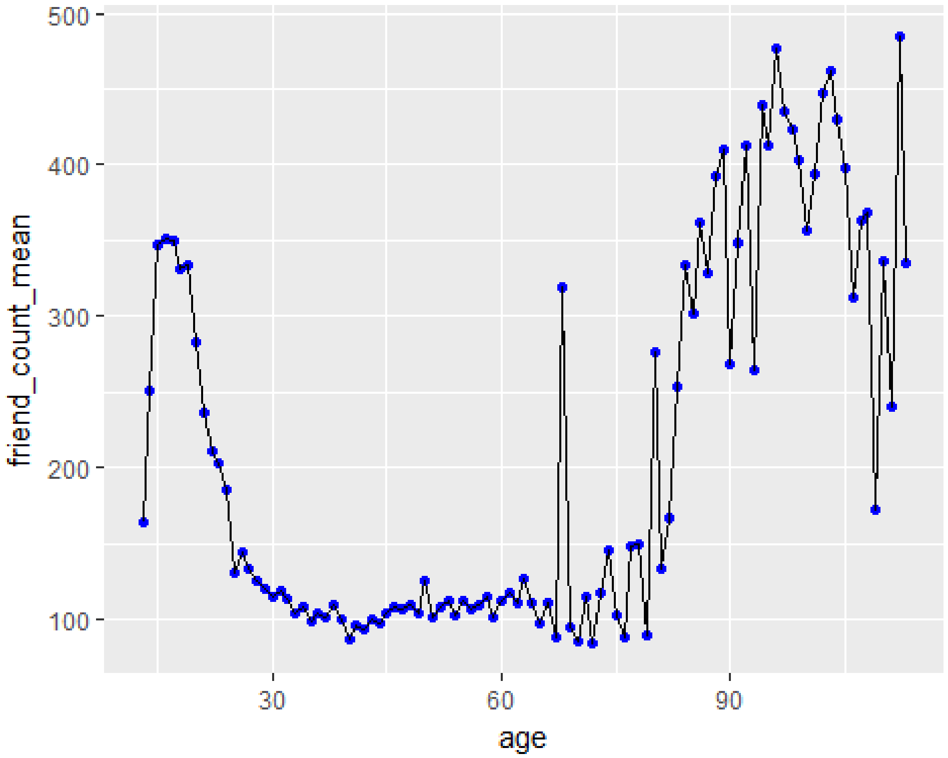
```
## Warning: Removed 5157 rows containing missing values (geom_point).
```



```
ggplot(aes(x=age, y=friendships_initiated, color='blue'), data=pf)+
  geom_jitter(alpha = 1/10, position = position_jitter(h = 0))+
  coord_trans(y='sqrt')
```

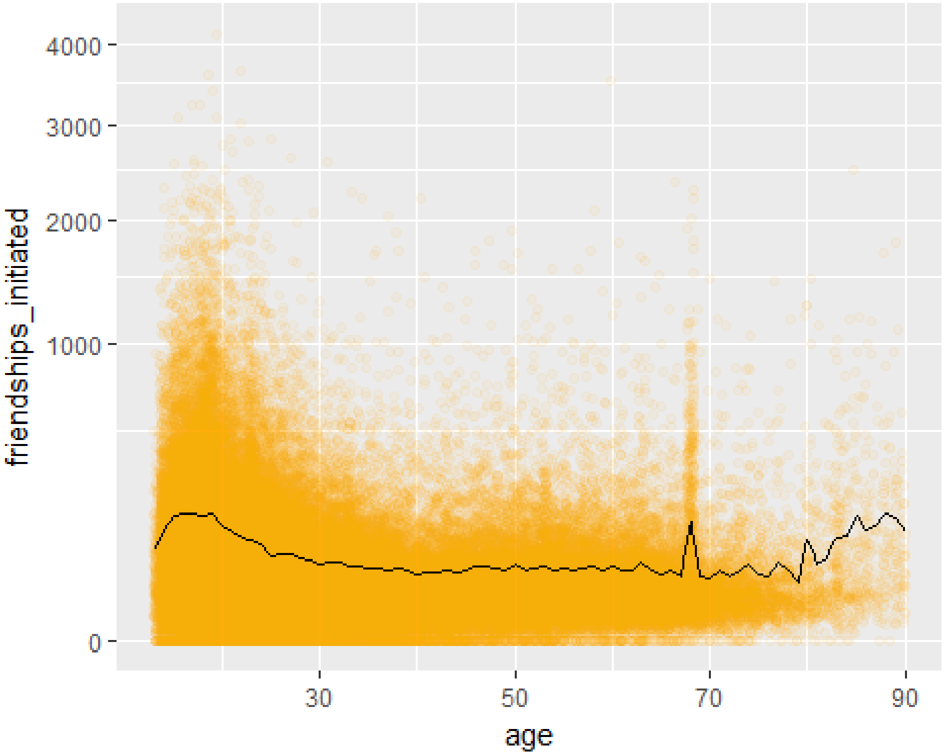


```
#Arranging by age group. The %>% chains the function to the pf dataset
pf.fc_by_age <- pf %>%
  group_by(age) %>%
  summarise(friend_count_mean = mean(friend_count),
            friend_count_median = median(friend_count),
            n = n()) %>%
  arrange(age)

#Plotting the median friend count
ggplot(aes(x=age, y=friend_count_mean), data = pf.fc_by_age)+
  geom_point(color = 'blue') + geom_line()
```



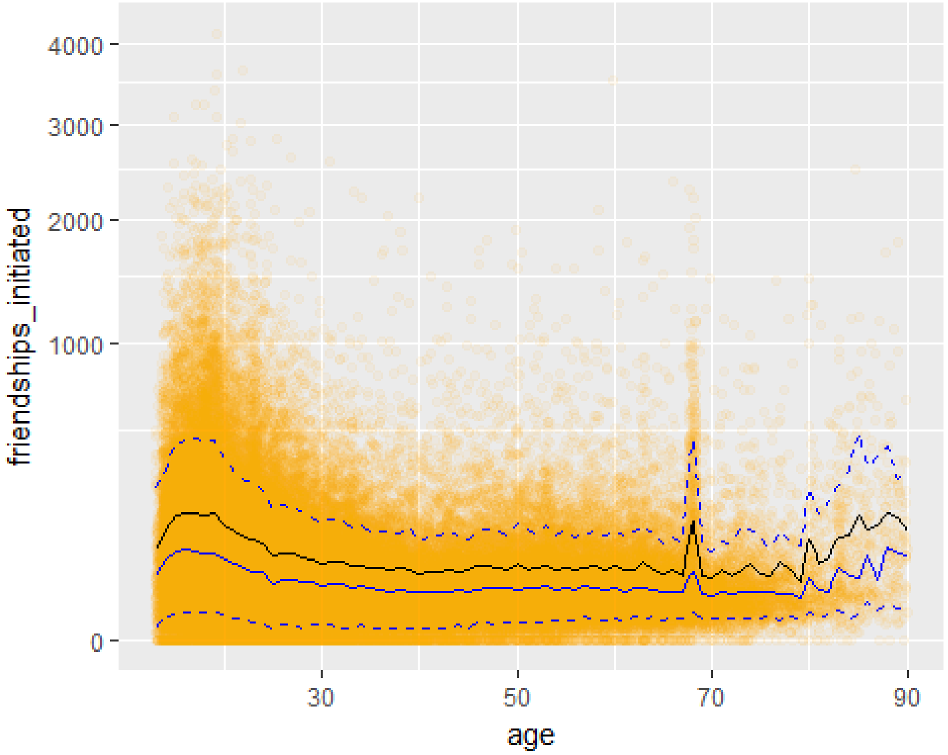
```r
ggplot(aes(x=age, y=friendships_initiated), data=pf)+
  xlim(13,90) +
  geom_jitter(alpha = 0.05,
              position = position_jitter(h = 0),
              color = 'orange') +
  coord_trans(y='sqrt')+
  geom_line(stat = 'summary', fun.y=mean)
```

```
## Warning: Removed 4906 rows containing non-finite values (stat_summary).

## Warning: Removed 5191 rows containing missing values (geom_point).
```

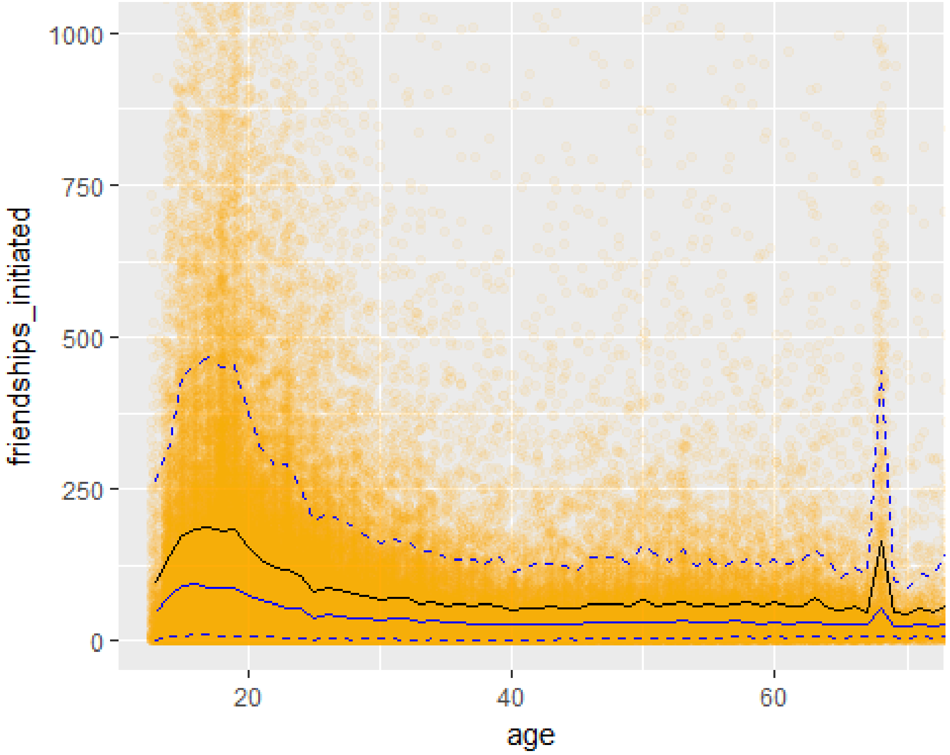


```r
#Below we added the have the 90% and 10% quantiles (dotted blue lines)
#The mean is the black line
#The median is the solid blue line
ggplot(aes(x=age, y=friendships_initiated), data=pf)+
  xlim(13,90) +
  geom_jitter(alpha = 0.05,
              position = position_jitter(h = 0),
              color = 'orange') +
  coord_trans(y='sqrt')+
  geom_line(stat = 'summary', fun.y=mean) +
  geom_line(stat = 'summary', fun.y=quantile, fun.args = list(probs = .1), li
netype = 2, color = 'blue')+
  geom_line(stat = 'summary', fun.y=quantile, fun.args = list(probs = .5), co
lor = 'blue')+
  geom_line(stat = 'summary', fun.y=quantile, fun.args = list(probs = .9), li
netype = 2, color = 'blue')
```

```
## Warning: Removed 4906 rows containing non-finite values (stat_summary).

## Warning: Removed 4906 rows containing non-finite values (stat_summary).

## Warning: Removed 4906 rows containing non-finite values (stat_summary).

## Warning: Removed 4906 rows containing non-finite values (stat_summary).

## Warning: Removed 5193 rows containing missing values (geom_point).
```



```r
#Using coord_cartesian to zoom in to 600 and below
ggplot(aes(x=age, y=friendships_initiated), data=pf)+
  geom_point(alpha = 0.05,
             position = position_jitter(h = 0),
             color = 'orange') +
  coord_cartesian(xlim = c(13,70), ylim = c(0,1000))+
  geom_line(stat = 'summary', fun.y=mean) +
  geom_line(stat = 'summary', fun.y=quantile, fun.args = list(probs = .1),
          linetype = 2, color = 'blue')+
  geom_line(stat = 'summary', fun.y=quantile, fun.args = list(probs = .5),
          color = 'blue')+
  geom_line(stat = 'summary', fun.y=quantile, fun.args = list(probs = .9),
          linetype = 2, color = 'blue')
```



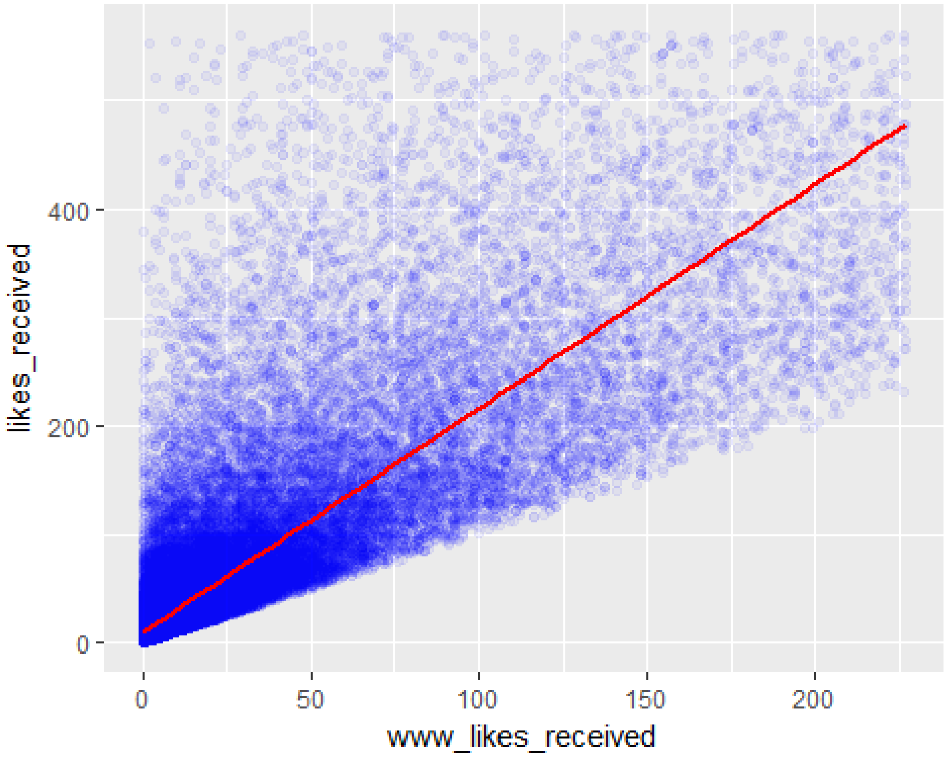
```r
#Calculating the correlation coefficient betweenage count and friend count 0.
3 = meaningful but small
#No real correlation
with(pf, cor.test(age, friend_count, method='pearson'))
```

```
##
##  Pearson's product-moment correlation
##
## data:  age and friend_count
## t = -8.6268, df = 99001, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.03363072 -0.02118189
## sample estimates:
##         cor
## -0.02740737
```

```r
#When we subset the data for users 70 and younger we can see negative correla
tion
with(subset(pf, age<=70), cor.test(age, friend_count))
```

```
##
##  Pearson's product-moment correlation
##
## data:  age and friend_count
## t = -52.592, df = 91029, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.1780220 -0.1654129
## sample estimates:
##        cor
## -0.1717245
```

```r
ggplot(pf, aes(x=www_likes_received, y=likes_received))+geom_point(alpha=0.05
, color = 'blue')+
  xlim(0, quantile(pf$www_likes_received, 0.95))+
  ylim(0, quantile(pf$likes_received, 0.95))+
  geom_smooth(method = 'lm', color = 'red')

## Warning: Removed 6075 rows containing non-finite values (stat_smooth).

## Warning: Removed 6075 rows containing missing values (geom_point).
```

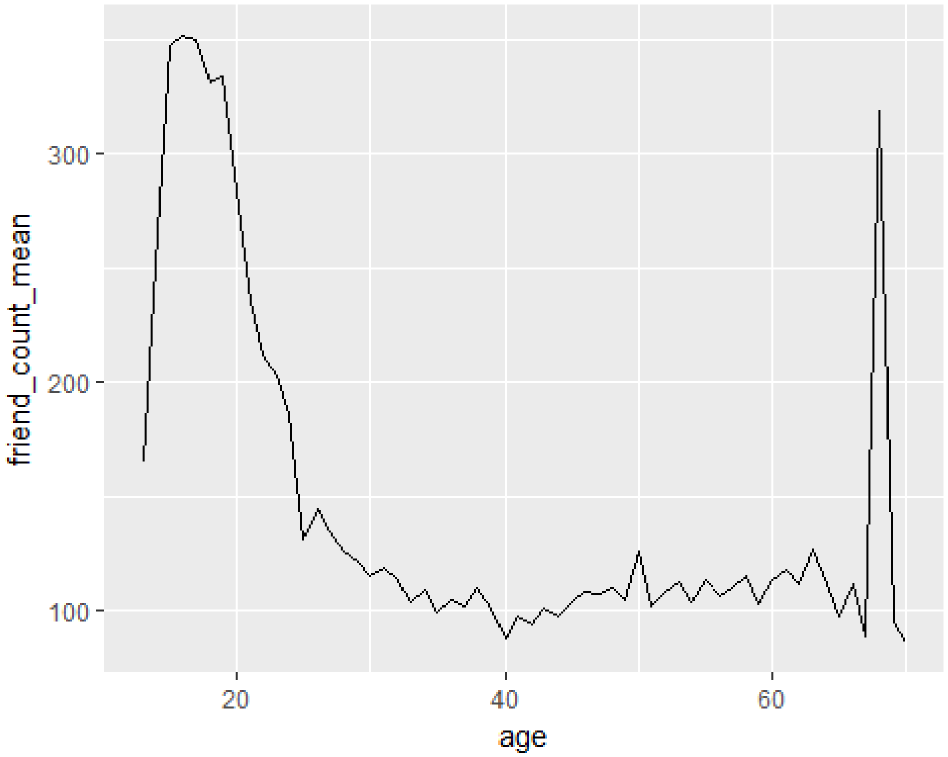


```r
with(pf, cor.test(www_likes_received, likes_received))

## 
##  Pearson's product-moment correlation
## 
## data:  www_likes_received and likes_received
## t = 937.1, df = 99001, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9473553 0.9486176
## sample estimates:
##       cor 
## 0.9479902

#Adding a category that also factors in months to the age
pf$age_with_months <- with(pf, age + (12 - dob_month)/12)
```

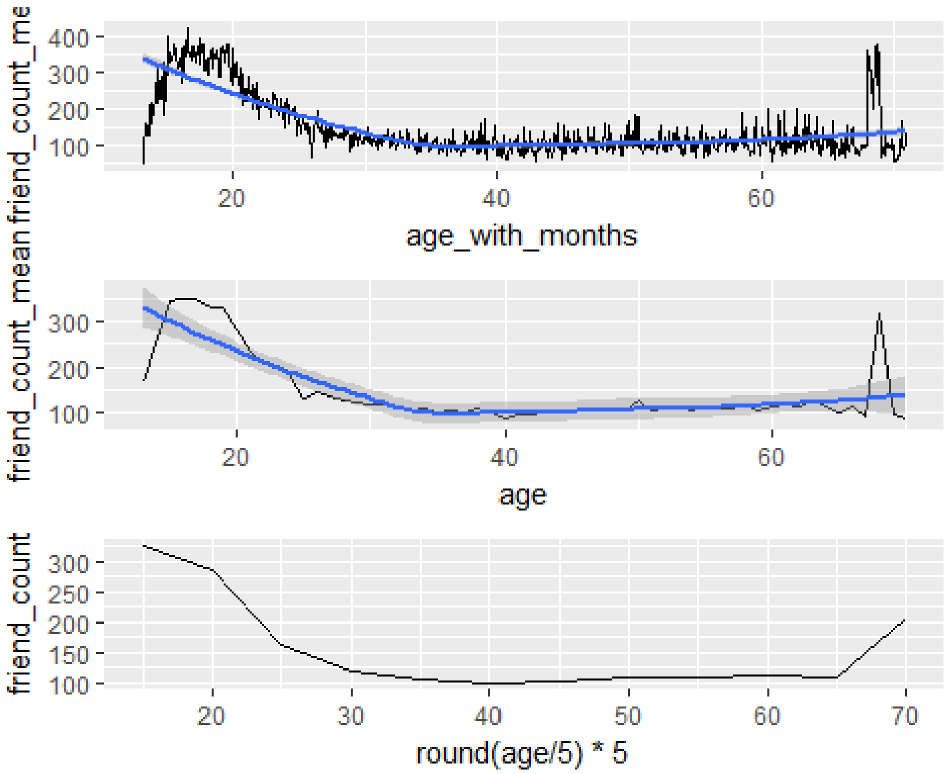
```r
library(dplyr)
pf.fc_by_age_months <- pf %>%
  group_by(age_with_months) %>%
  summarise(friend_count_mean = mean(friend_count),
            friend_count_median = median(friend_count),
            n = n()) %>%
  arrange(age_with_months)

head(pf.fc_by_age_months)
```

```
## # A tibble: 6 x 4
##   age_with_months friend_count_mean friend_count_median     n
##             <dbl>             <dbl>               <dbl> <int>
## 1        13.16667          46.33333                30.5     6
## 2        13.25000         115.07143                23.5    14
## 3        13.33333         136.20000                44.0    25
## 4        13.41667         164.24242                72.0    33
## 5        13.50000         131.17778                66.0    45
## 6        13.58333         156.81481                64.0    54
```

```r
ggplot(aes(x=age, y=friend_count_mean), data = subset(pf.fc_by_age, age < 71)
)+
  geom_line()
```



```
#plotting the 71 or less age with months
#+Age, then plotting the average so its a smooth rolling chart of 5yr avgs
#Note we want avg friend count so we apply the mean function to the line

p1 <- ggplot(subset(pf.fc_by_age_months, age_with_months < 71), aes(x=age_wit
h_months, y=friend_count_mean))+            geom_line()+
  geom_smooth()

p2 <- ggplot(aes(x=age, y=friend_count_mean), data = subset(pf.fc_by_age, age
< 71))+
  geom_line()+
  geom_smooth()

p3 <- ggplot(subset(pf, age<71), aes(x=round(age/5)*5, y=friend_count)) +
  geom_line(stat = 'summary', fun.y = mean)

library(gridExtra)

grid.arrange(p1, p2, p3, ncol = 1)

## `geom_smooth()` using method = 'loess'

## `geom_smooth()` using method = 'loess'
```


friend_count_me
age_with_months
friend_count_mean
age
friend_count
round(age/5) * 5