Lesson 5

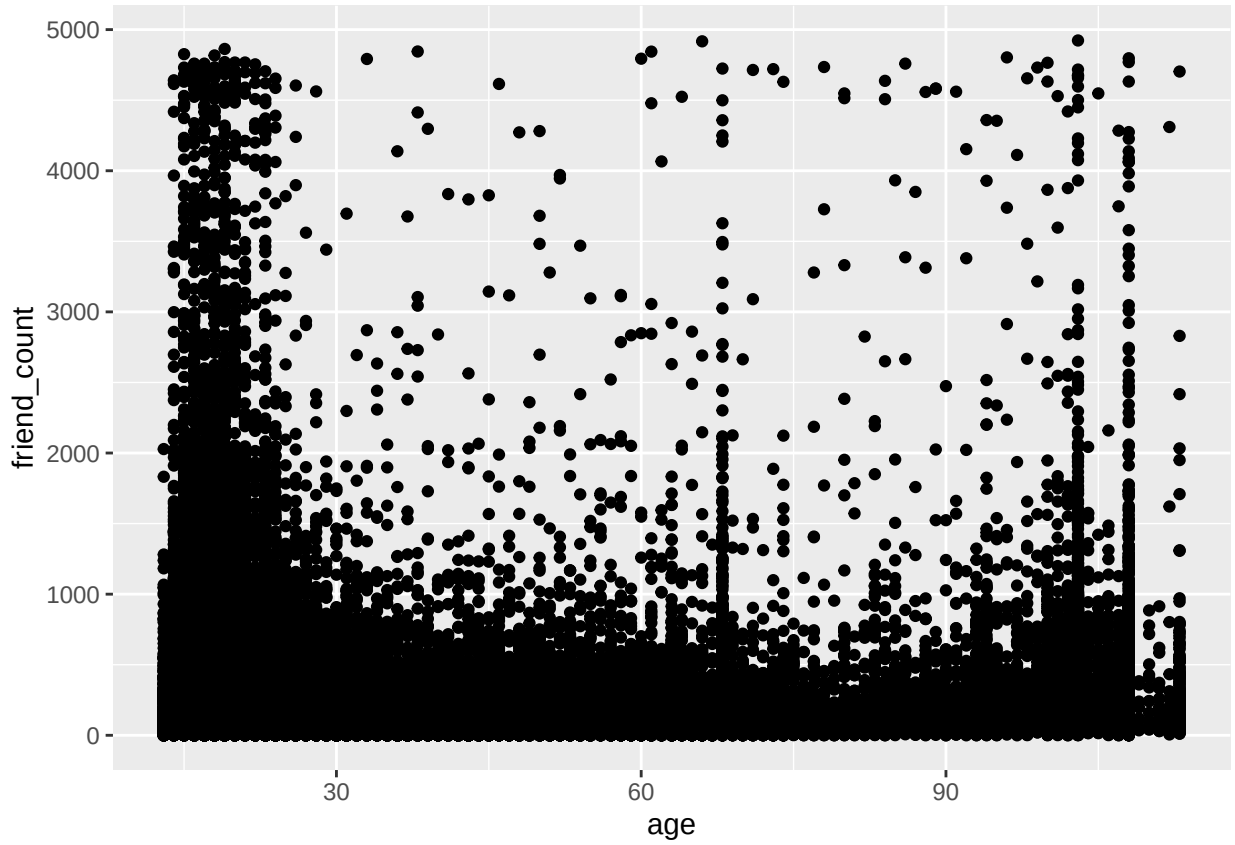
Scatterplots and Perceived Audience Size

Notes:

Scatterplots

Notes:

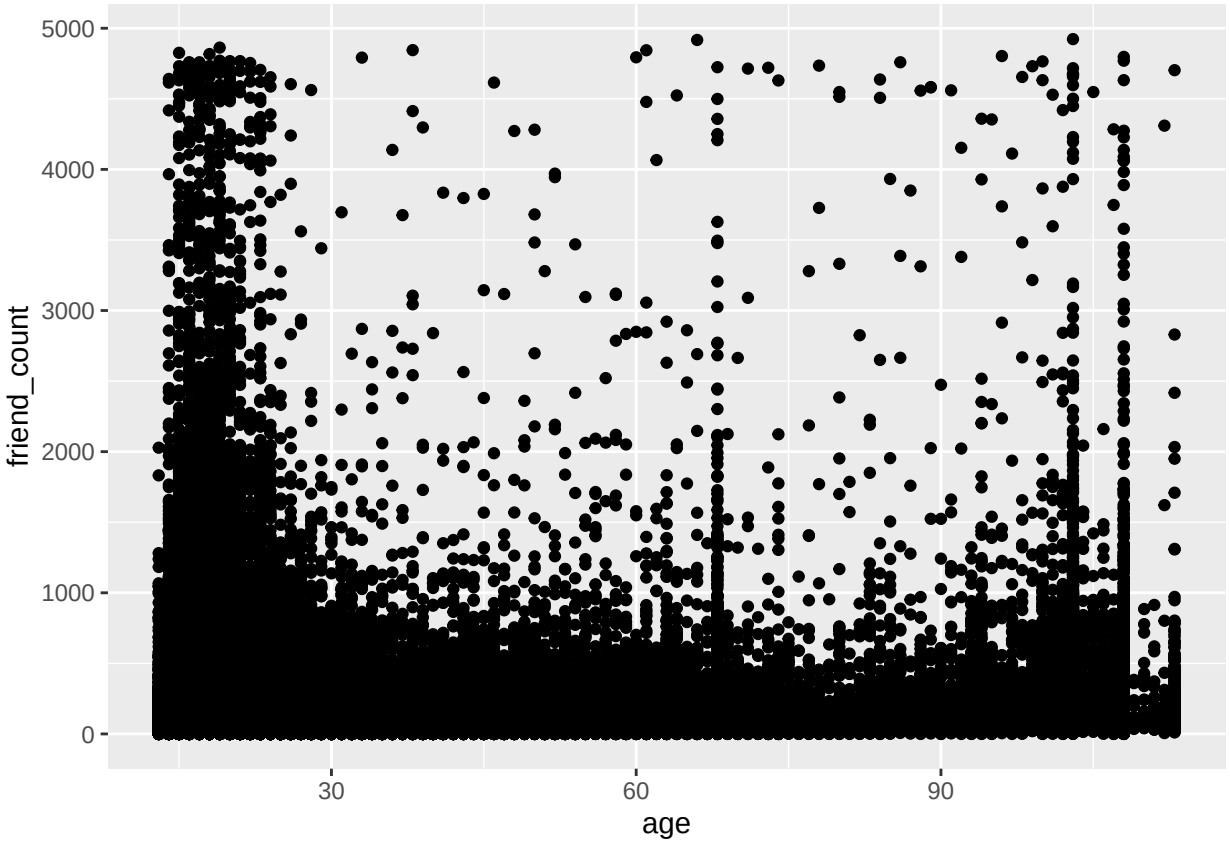
```
library(ggplot2)
pf=read.csv('pseudo_facebook.tsv', sep = '\t')

qplot(x = age, y = friend_count, data = pf)
```




```
ggsave('scatter1.png')
```

```
## Saving 6.5 x 4.5 in image
```

```
qplot(age, friend_count, data = pf)
```




```
ggsave('scatter2.png')
```

```
## Saving 6.5 x 4.5 in image
```

What are some things that you notice right away?

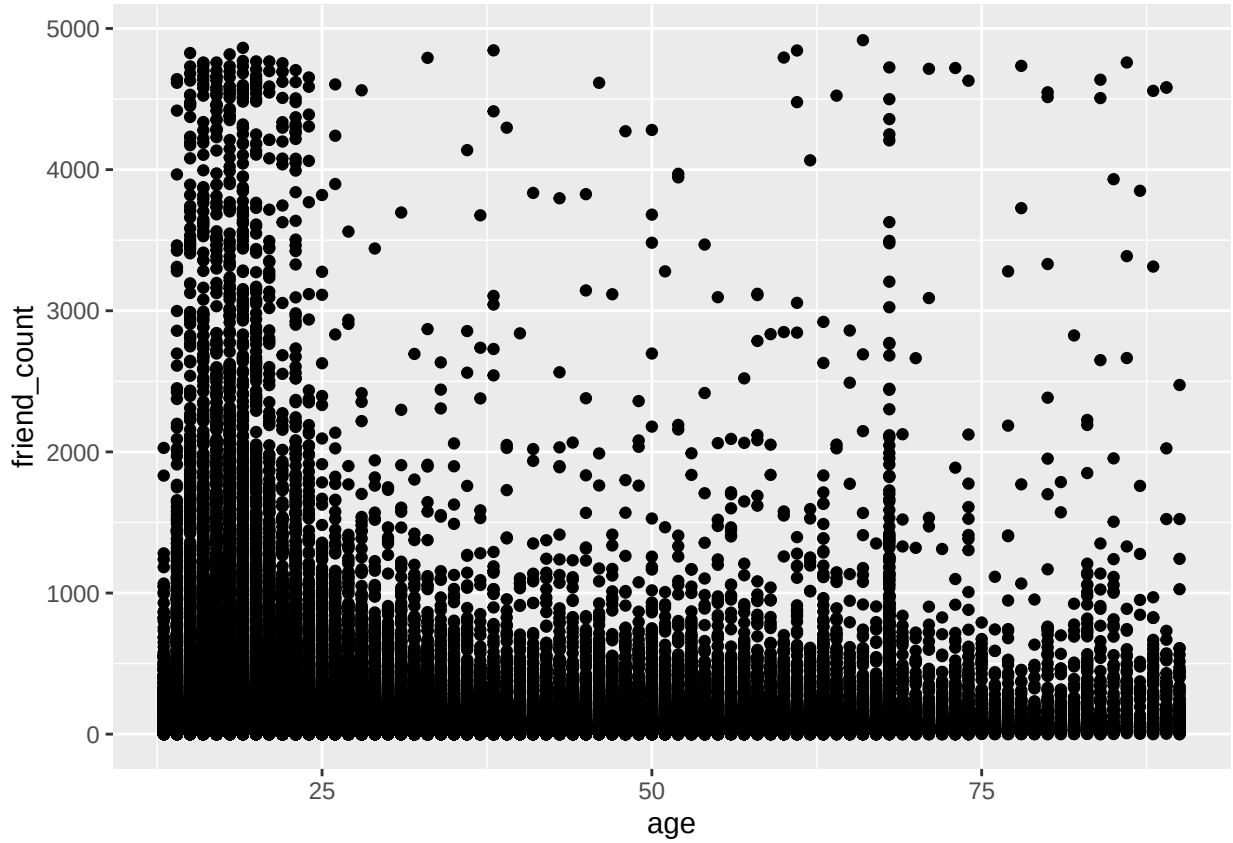
Response: range around25, much more friend_count. ***

ggplot Syntax

Notes:

```
ggplot(aes(x = age, y = friend_count), data = pf) + geom_point() +
  xlim(13, 90)
```

```
## Warning: Removed 4906 rows containing missing values (geom_point).
```




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

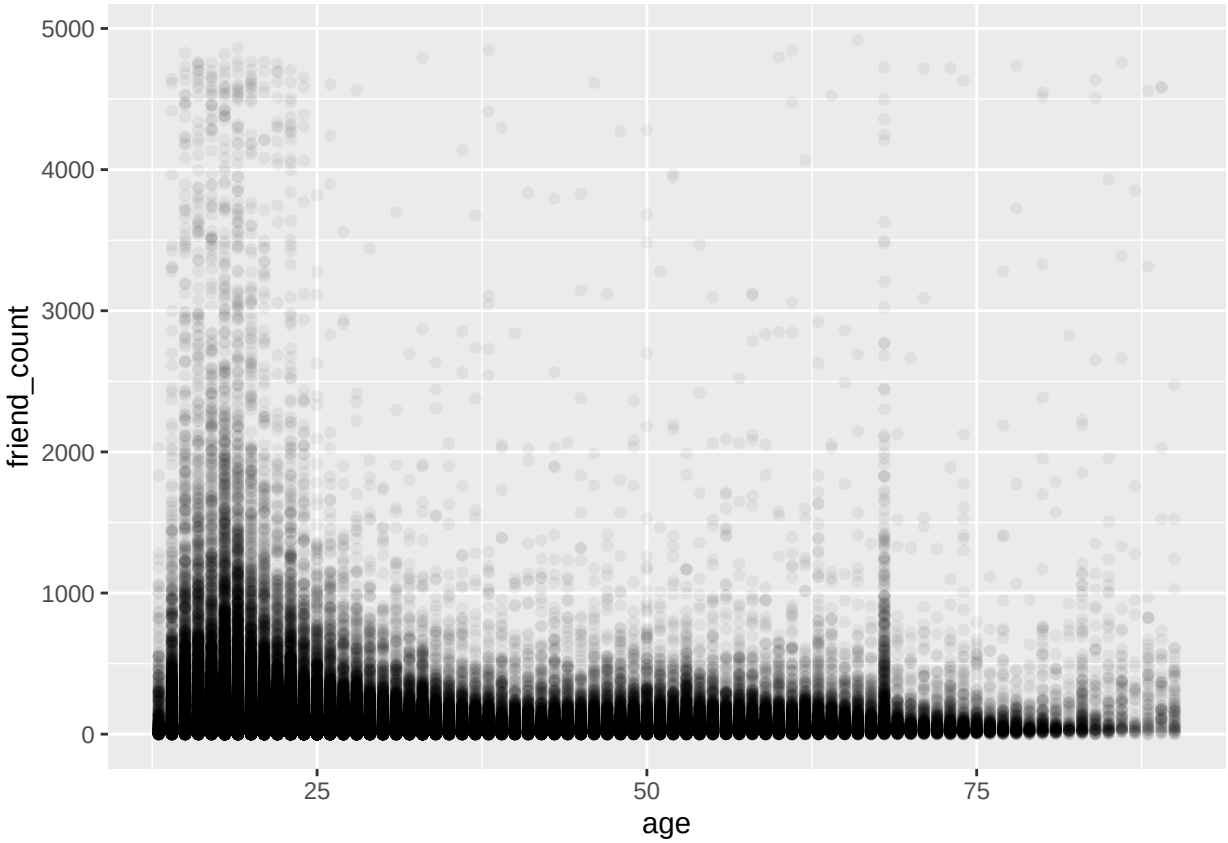
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   13.00   20.00   28.00   37.28   50.00  113.00
```

Overplotting

Notes:

```
ggplot(aes(x = age, y = friend_count), data = pf) +
  geom_point(alpha = 1/20) +
  xlim(13, 90)
```

```
## Warning: Removed 4906 rows containing missing values (geom_point).
```



What do you notice in the plot?

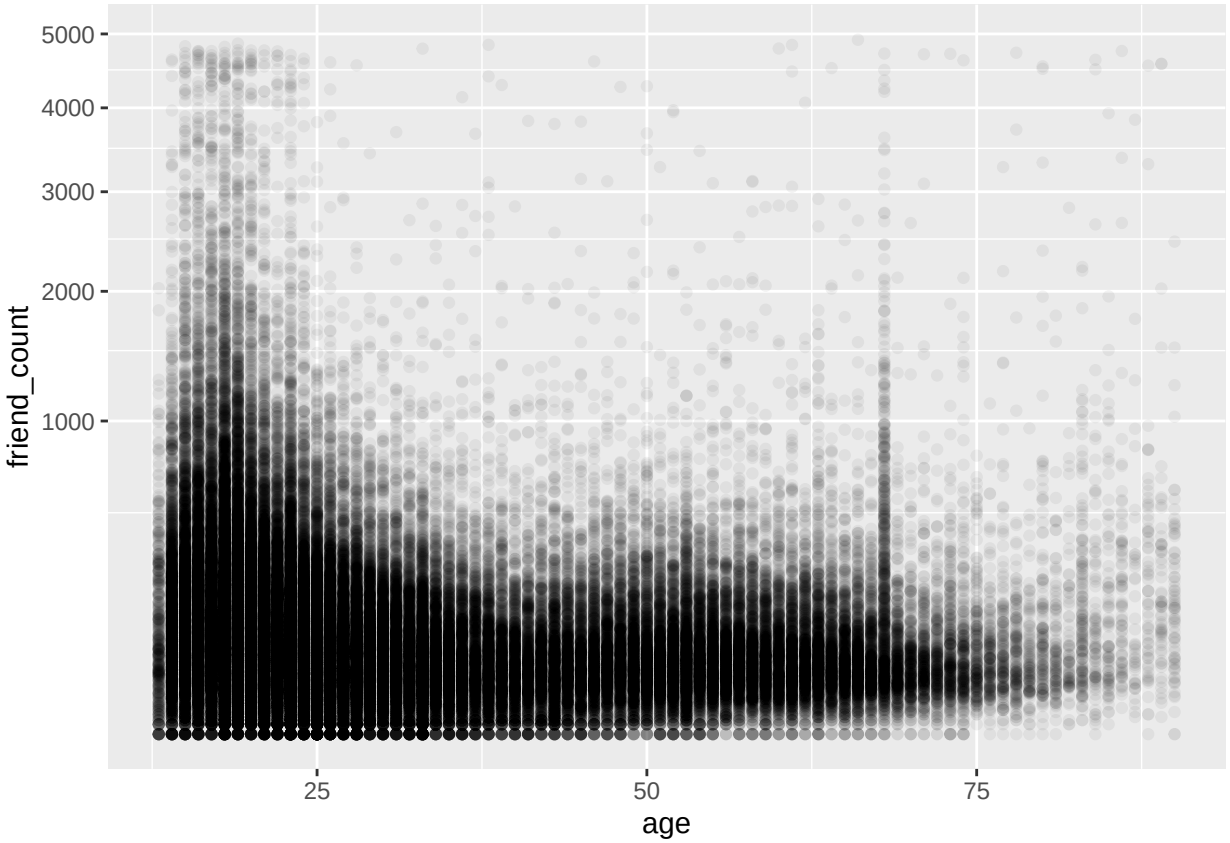
Response: 1. friend_count of more younger is below 1000; 2. y-axis is not suitable. ***

Coord_trans()

Notes:

```
ggplot(aes(x = age, y = friend_count), data = pf) +
  geom_point(alpha=1/20) +
  xlim(13, 90) +
  coord_trans(y = "sqrt")
```

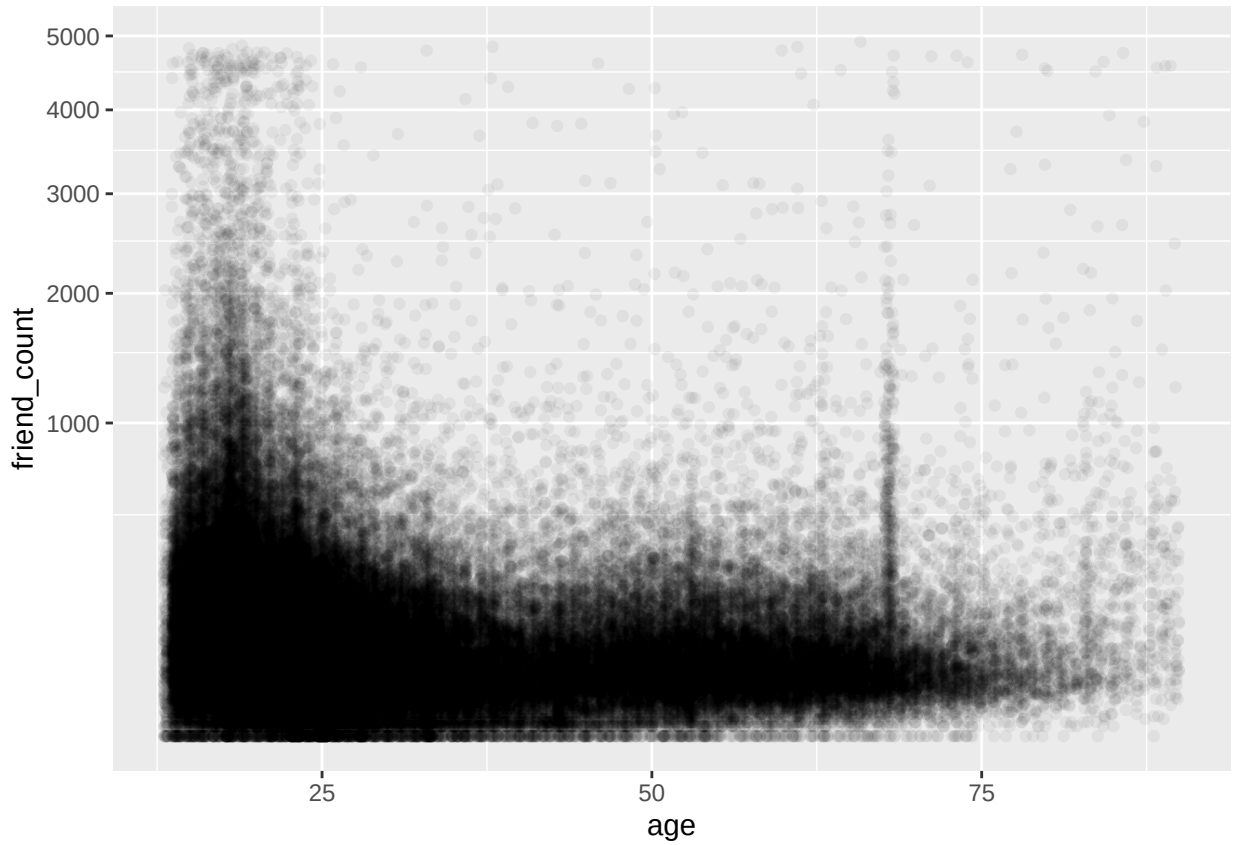
```
## Warning: Removed 4906 rows containing missing values (geom_point).
```



Look up the documentation for coord_trans() and add a layer to the plot that transforms friend_count using the square root function. Create your plot!

```
ggplot(aes(x = age, y = friend_count), data = pf) +
  geom_point(alpha=1/20, position = position_jitter(h = 0)) +
  xlim(13, 90) +
  coord_trans(y = "sqrt")
```

```
## Warning: Removed 5183 rows containing missing values (geom_point).
```



What do you notice?

The density for the left corner is much higher.

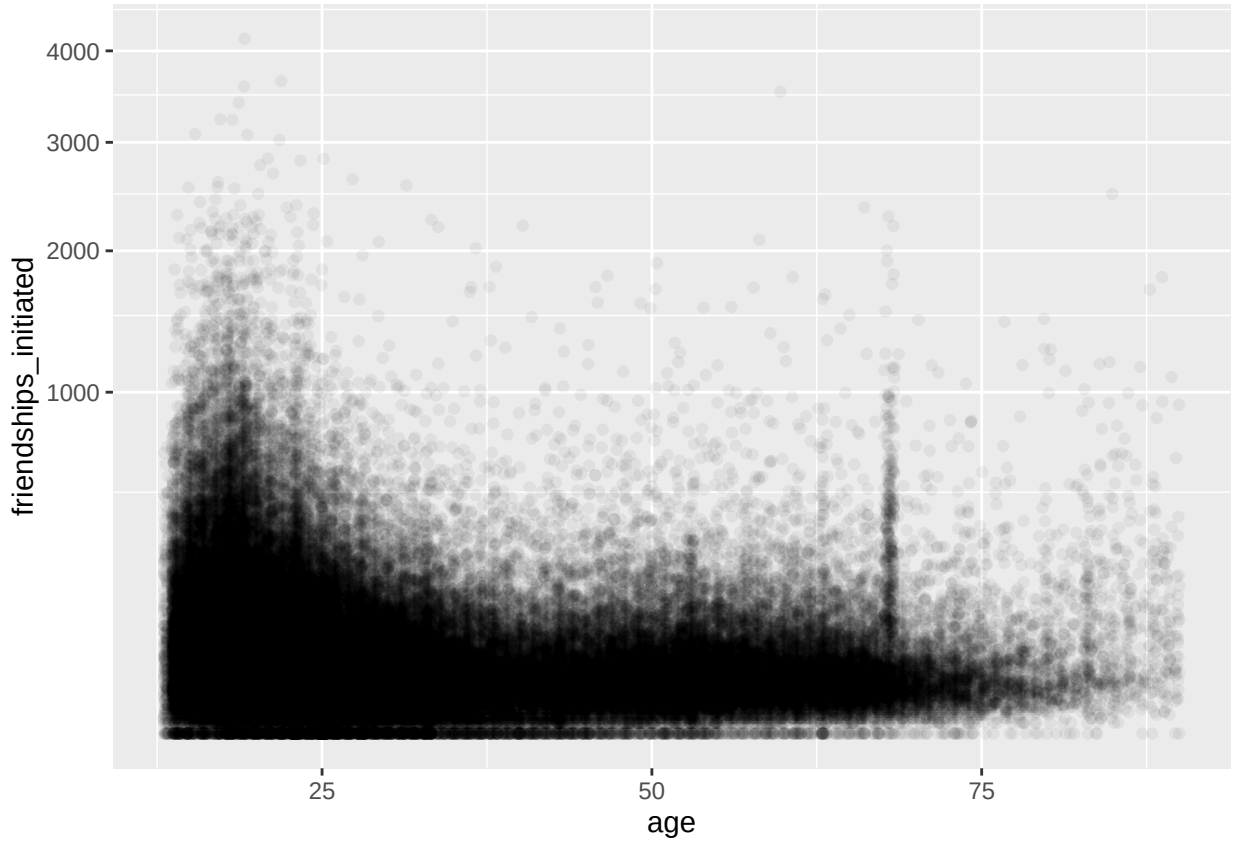
Alpha and Jitter

Notes:

```
# Examine the relationship between
# friendships_initiated (y) and age (x)
# using the ggplot syntax.

# We recommend creating a basic scatter
# plot first to see what the distribution looks like.
# and then adjusting it by adding one layer at a time.

# What are your observations about your final plot?

# Remember to make adjustments to the breaks
# of the x-axis and to use apply alpha and jitter.

# ENTER ALL OF YOUR CODE FOR YOUR PLOT BELOW THIS LINE.
```

```
# ============================================================
ggplot(aes(x = age, y = friendships_initiated), data = pf) +
  geom_point(alpha=1/20, position = position_jitter(h=0)) +
  xlim(13,90) +
  coord_trans(y = "sqrt")
```

```
## Warning: Removed 5184 rows containing missing values (geom_point).
```




```
summary(pf$friendships_initiated)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0    17.0    46.0   107.5   117.0  4144.0
```

Overplotting and Domain Knowledge

Notes:

Conditional Means

Notes:

```
#install.packages("dplyr")
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
```

```
age_groups = group_by(pf, age)
pf.fc_by_age = summarise(age_groups,
          friend_count_mean = mean(friend_count),
          friend_count_median = median(friend_count),
          n = n())
pf.fc_by_age = arrange(pf.fc_by_age, age)
head(pf.fc_by_age)
```

```
## # A tibble: 6 x 4
##     age friend_count_mean friend_count_median     n
##   <int>             <dbl>               <dbl> <int>
## 1    13              165.                74    484
## 2    14              251.               132   1925
## 3    15              348.               161   2618
## 4    16              352.               172.  3086
## 5    17              350.               156   3283
## 6    18              331.               162   5196
```

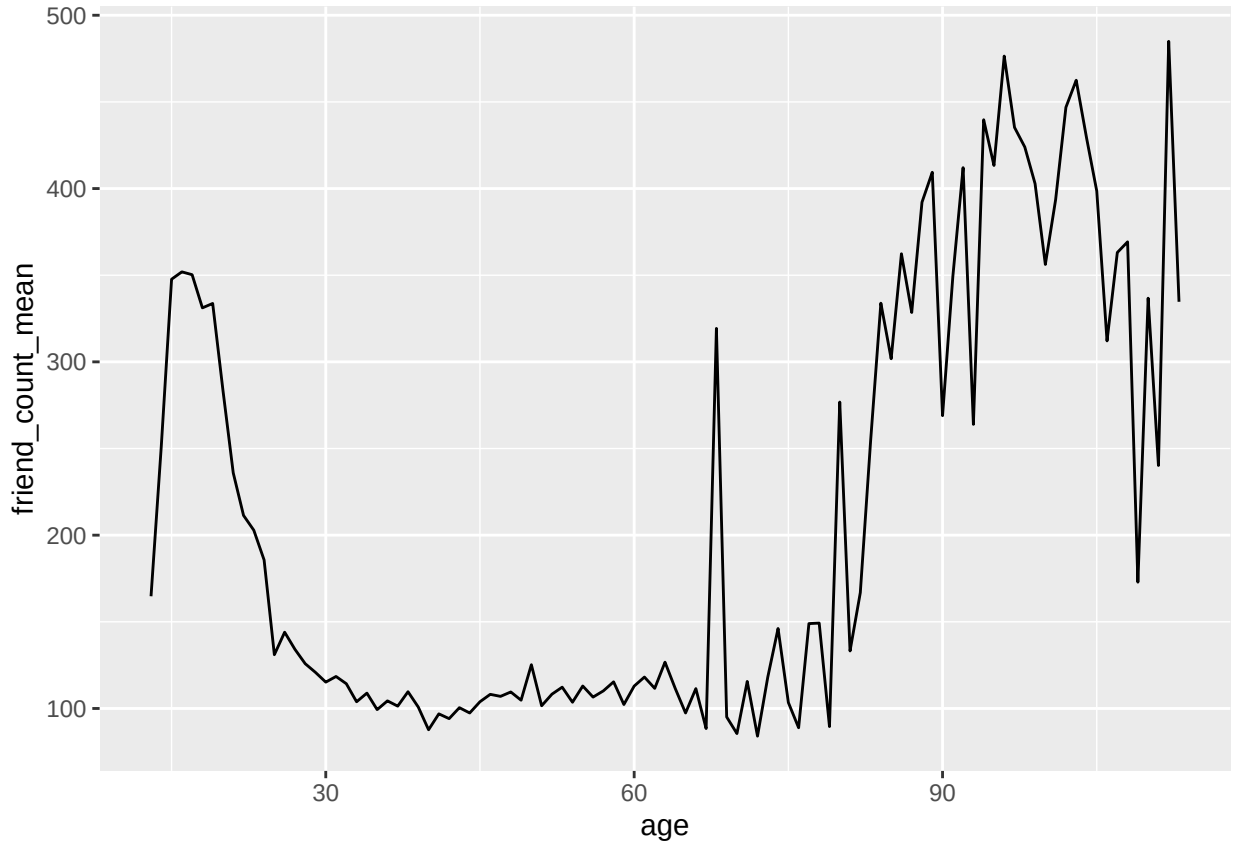
Create your plot!

```
library(dplyr)
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)

head(pf.fc_by_age,15)
```

```
## # A tibble: 15 x 4
##      age friend_count_mean friend_count_median     n
##    <int>             <dbl>               <dbl> <int>
```

```
##  1    13              165.                74    484
##  2    14              251.               132   1925
##  3    15              348.               161   2618
##  4    16              352.               172.  3086
##  5    17              350.               156   3283
##  6    18              331.               162   5196
##  7    19              334.               157   4391
##  8    20              283.               135   3769
##  9    21              236.               121   3671
## 10    22              211.               106   3032
## 11    23              203.                93   4404
## 12    24              186.                92   2827
## 13    25              131.                62   3641
## 14    26              144.                75   2815
## 15    27              134.                72   2240
```

```r
# Plot mean friend count vs. age using a line graph.
# Be sure you use the correct variable names
# and the correct data frame. You should be working
# with the new data frame created from the dplyr
# functions. The data frame is called 'pf.fc_by_age'.

# Use geom_line() rather than geom_point to create
# the plot. You can look up the documentation for
# geom_line() to see what it does.

# ENTER ALL OF YOUR CODE TO CREATE THE PLOT BELOW THIS LINE.
# ============================================================
library(ggplot2)
ggplot(aes(x = age, y = friend_count_mean), data = pf.fc_by_age) +
  geom_line()
```

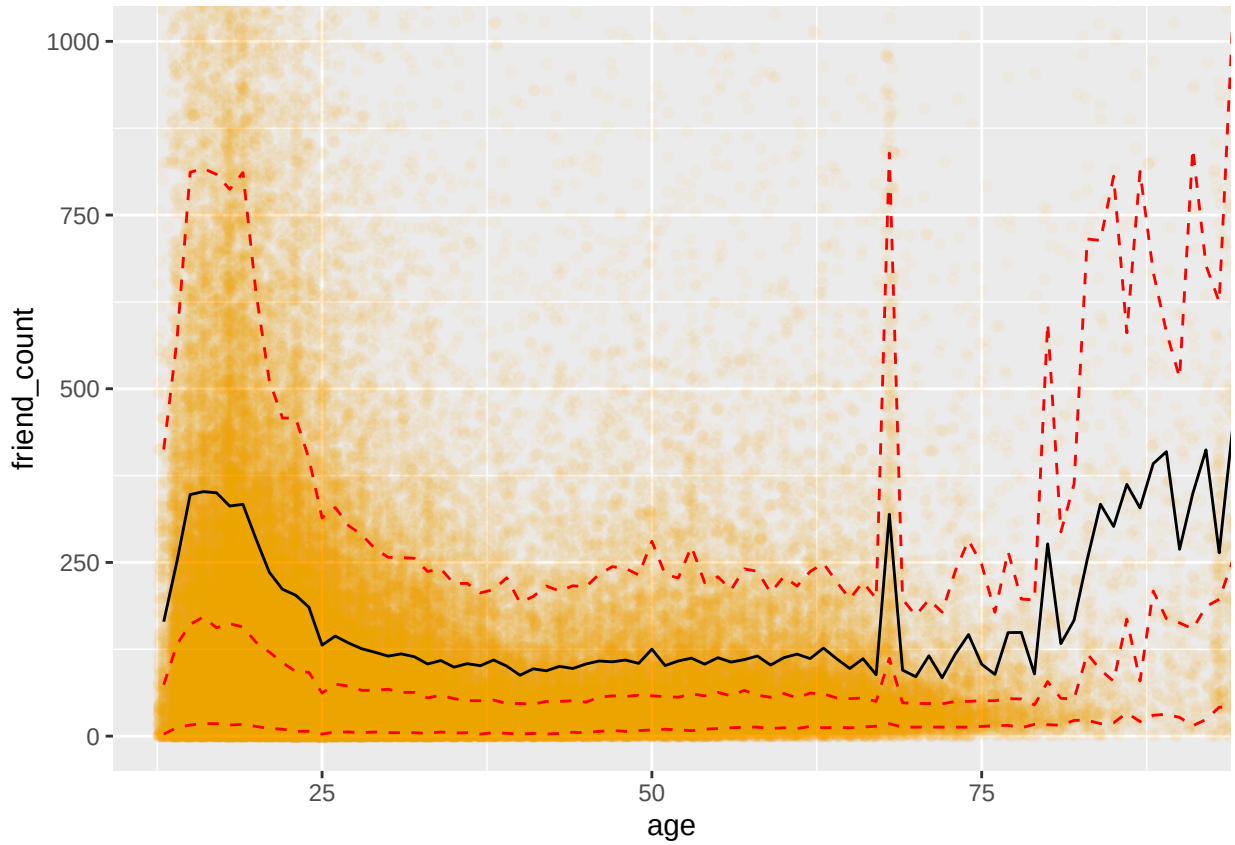


Overlaying Summaries with Raw Data

Notes:

```
library(dplyr)
library(Lahman)
ggplot(aes(x = age, y = friend_count), data = pf) +
  coord_cartesian(xlim = c(13, 90), ylim = c(0, 1000)) +
  geom_point(alpha = 1/20,
             position = position_jitter(h = 0),
             color = 'orange') +
  stat_summary(fun.y = mean, geom = "line") +
  stat_summary(fun.y = quantile, fun.args = list(probs = .1), geom = "line", color = "red", linetype = 1
  stat_summary(fun.y = quantile, fun.args = list(probs = .5), geom = "line", color = "red", linetype = 1
  stat_summary(fun.y = quantile, fun.args = list(probs = .9), geom = "line", color = "red", linetype = 1
```

```
## Warning: `fun.y` is deprecated. Use `fun` instead.

## Warning: `fun.y` is deprecated. Use `fun` instead.

## Warning: `fun.y` is deprecated. Use `fun` instead.

## Warning: `fun.y` is deprecated. Use `fun` instead.
```



What are some of your observations of the plot?

Response: plot of them are almost same. ***

Moira: Histogram Summary and Scatterplot

See the Instructor Notes of this video to download Moira's paper on perceived audience size and to see the final plot.

Notes:

Correlation

Notes:

```
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
```

Look up the documentation for the cor.test function.

What's the correlation between age and friend count? Round to three decimal places. Response:

Correlation on Subsets

Notes:

```
with(subset(pf, 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
```

Correlation Methods

Notes:

```
with(subset(pf, age <= 70), cor.test(age, friend_count), method = 'spearman')
```

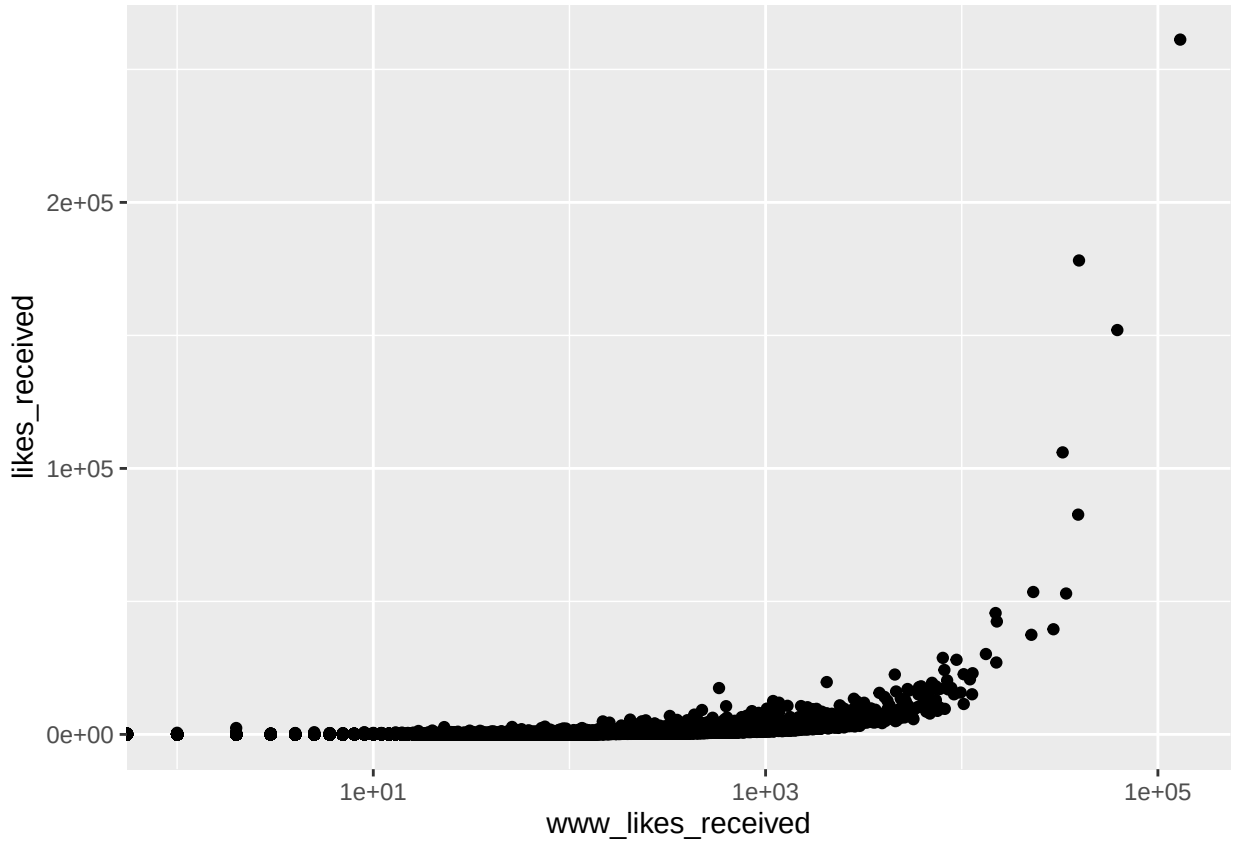
```
##
##  Pearson's product-moment correlation
##
## data:  age and friend_count
## t = -52.592, df = 91029, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.1780220 -0.1654129
## sample estimates:
##        cor
## -0.1717245
```

Create Scatterplots

Notes:

```
# Create a scatterplot of likes_received (y)
# vs. www_likes_received (x). Use any of the
# techniques that you've learned so far to
# modify the plot.

# ENTER ALL OF YOUR CODE TO CREATE THE PLOT BELOW THIS LINE.
# ============================================================
ggplot(aes(x = www_likes_received, y = likes_received), data = pf) +
  geom_point() +
  scale_x_log10()
```

```
## Warning: Transformation introduced infinite values in continuous x-axis
```



Strong Correlations

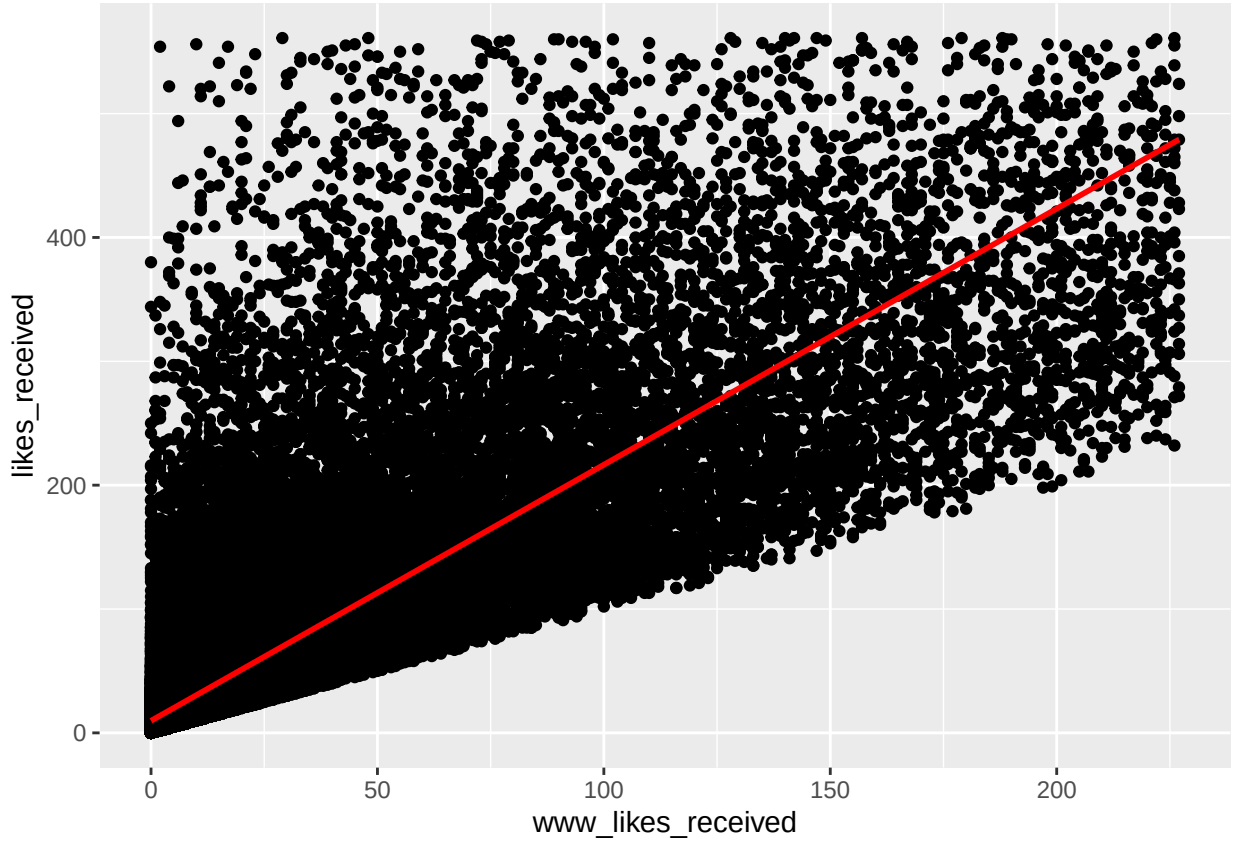
Notes:

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

```
## `geom_smooth()` using formula 'y ~ x'
```

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

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



What's the correlation betwen the two variables? Include the top 5% of values for the variable in the calculation and round to 3 decimal places.

```
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
```

Response:

Moira on Correlation

Notes:

More Caution with Correlation

Notes:

```
#install.packages('alr3')
library(alr3)
```

```
## Loading required package: car
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'carData'
```

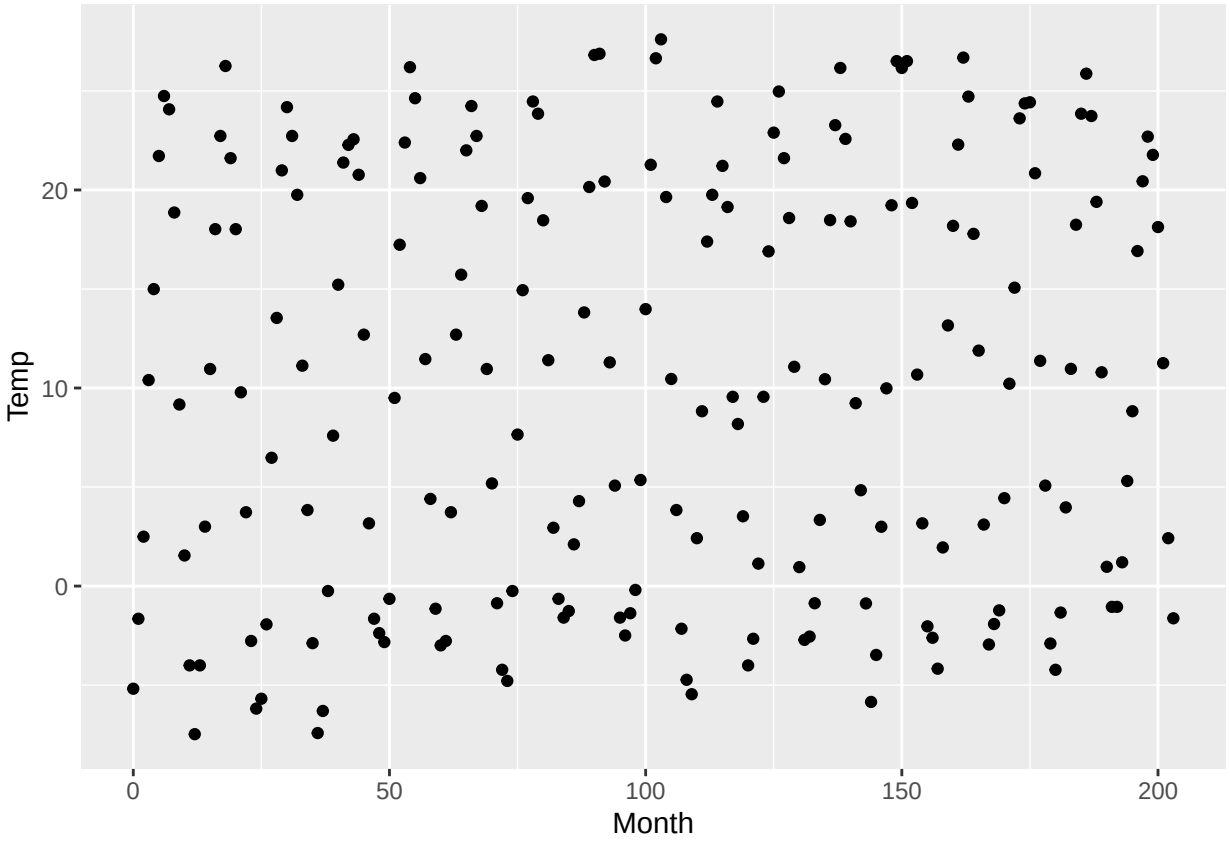
```
## The following object is masked from 'package:Lahman':
##
##     Salaries
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

Create your plot!

```
# Create a scatterplot of temperature (Temp)
# vs. months (Month).

# ENTER ALL OF YOUR CODE TO CREATE THE PLOT BELOW THIS LINE.
# ==========================================================
library(alr3)
data("Mitchell")
ggplot(aes(x = Month, y = Temp), data = Mitchell) +
  geom_point()
```



Noisy Scatterplots

a. Take a guess for the correlation coefficient for the scatterplot. 0
b. What is the actual correlation of the two variables? (Round to the thousandths place)

```
with(Mitchell, cor.test(Month, Temp))
```

```
##
##  Pearson's product-moment correlation
##
## data:  Month and Temp
## t = 0.81816, df = 202, p-value = 0.4142
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.08053637  0.19331562
## sample estimates:
##        cor
## 0.05747063
```

Making Sense of Data

Notes:

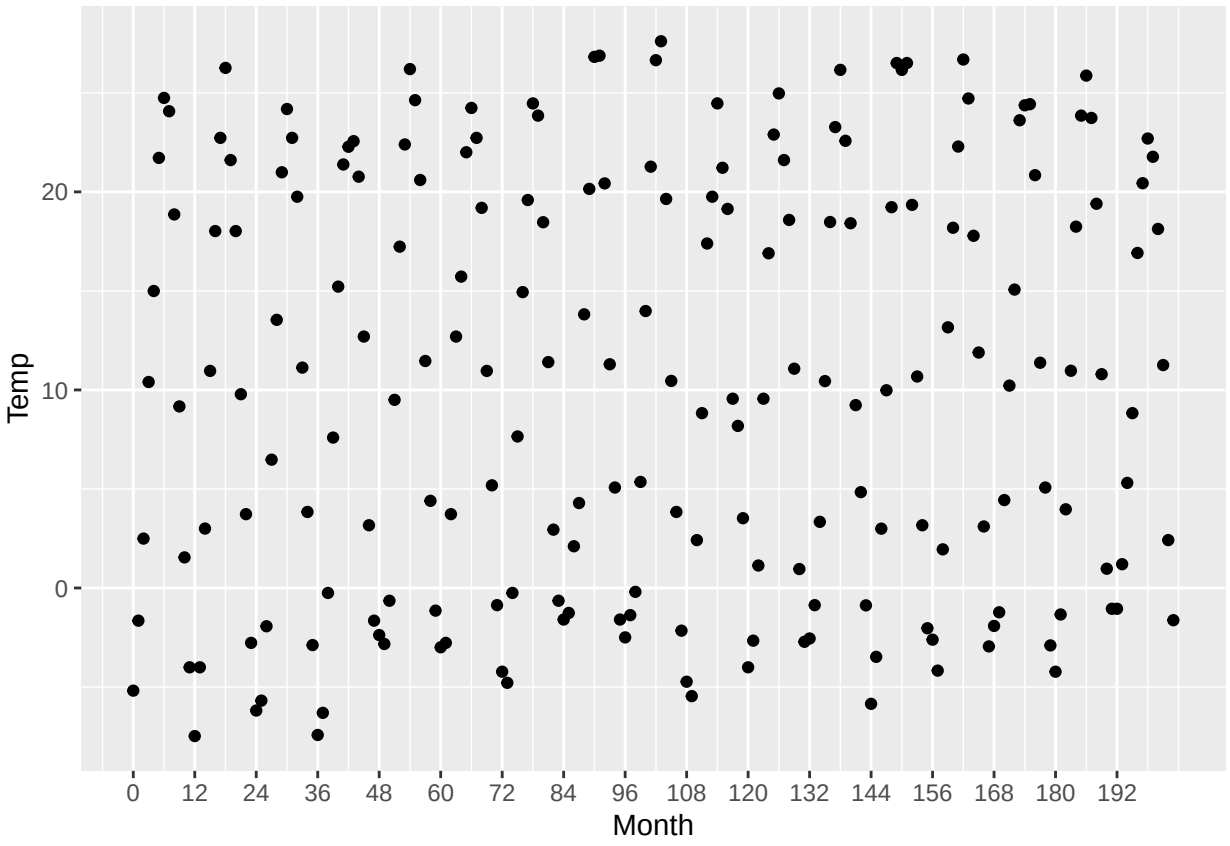
```
summary(Mitchell$Month)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   50.75  101.50  101.50  152.25  203.00
```

```
range(Mitchell$Month)
```

```
## [1]   0 203
```

```
ggplot(aes(x = Month, y = Temp), data = Mitchell) +
  geom_point()+
  scale_x_continuous(breaks=seq(0, 203, 12))
```



A New Perspective

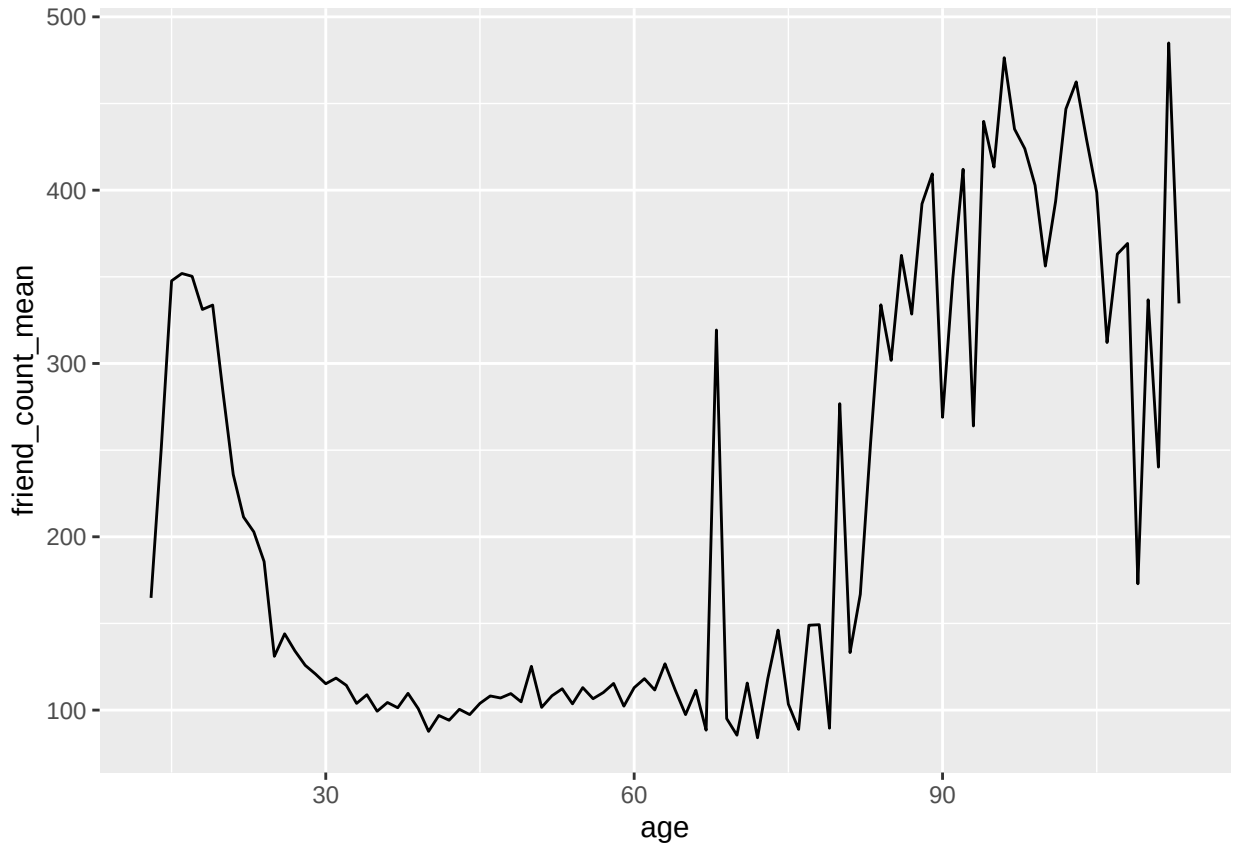
What do you notice? Response: there are some rule, like sin(),cos() function.

Watch the solution video and check out the Instructor Notes! Notes:

Understanding Noise: Age to Age Months

Notes:

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




```
head(pf.fc_by_age, 10)
```

```
## # A tibble: 10 x 4
##      age friend_count_mean friend_count_median     n
##    <int>             <dbl>                <dbl> <int>
##  1    13              165.                   74   484
##  2    14              251.                  132  1925
##  3    15              348.                  161  2618
##  4    16              352.                  172.  3086
##  5    17              350.                  156  3283
##  6    18              331.                  162  5196
##  7    19              334.                  157  4391
##  8    20              283.                  135  3769
##  9    21              236.                  121  3671
## 10    22              211.                  106  3032
```

```
pf.fc_by_age[17:19, ]
```

```
## # A tibble: 3 x 4
##     age friend_count_mean friend_count_median     n
##   <int>             <dbl>               <dbl> <int>
## 1    29              121.                66    1936
## 2    30              115.                67.5  1716
## 3    31              118.                63    1694
```

```
# Create a new variable, 'age_with_months', in the 'pf' data frame.
# Be sure to save the variable in the data frame rather than creating
# a separate, stand-alone variable. You will need to use the variables
# 'age' and 'dob_month' to create the variable 'age_with_months'.

# Assume the reference date for calculating age is December 31, 2013.

# This programming assignment WILL BE automatically graded. For
# this exercise, you need only create the 'age_with_months' variable;
# no further processing of the data frame is necessary.

# ENTER YOUR CODE BELOW THIS LINE
# ================================================================================

pf$age_with_months = pf$age + (12 - pf$dob_month) / 12
```

Age with Months Means

```
# Create a new data frame called
# pf.fc_by_age_months that contains
# the mean friend count, the median friend
# count, and the number of users in each
# group of age_with_months. The rows of the
# data framed should be arranged in increasing
# order by the age_with_months variable.

# For example, the first two rows of the resulting
# data frame would look something like...

# age_with_months  friend_count_mean    friend_count_median n
#              13            275.0000                  275 2
#        13.25000            133.2000                  101 11

# See the Instructor Notes for two hints if you get stuck.
# This programming assignment will automatically be graded.

suppressMessages(library(dplyr))

# ENTER YOUR CODE BELOW THIS LINE
# ================================================================================
```

```
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)
```

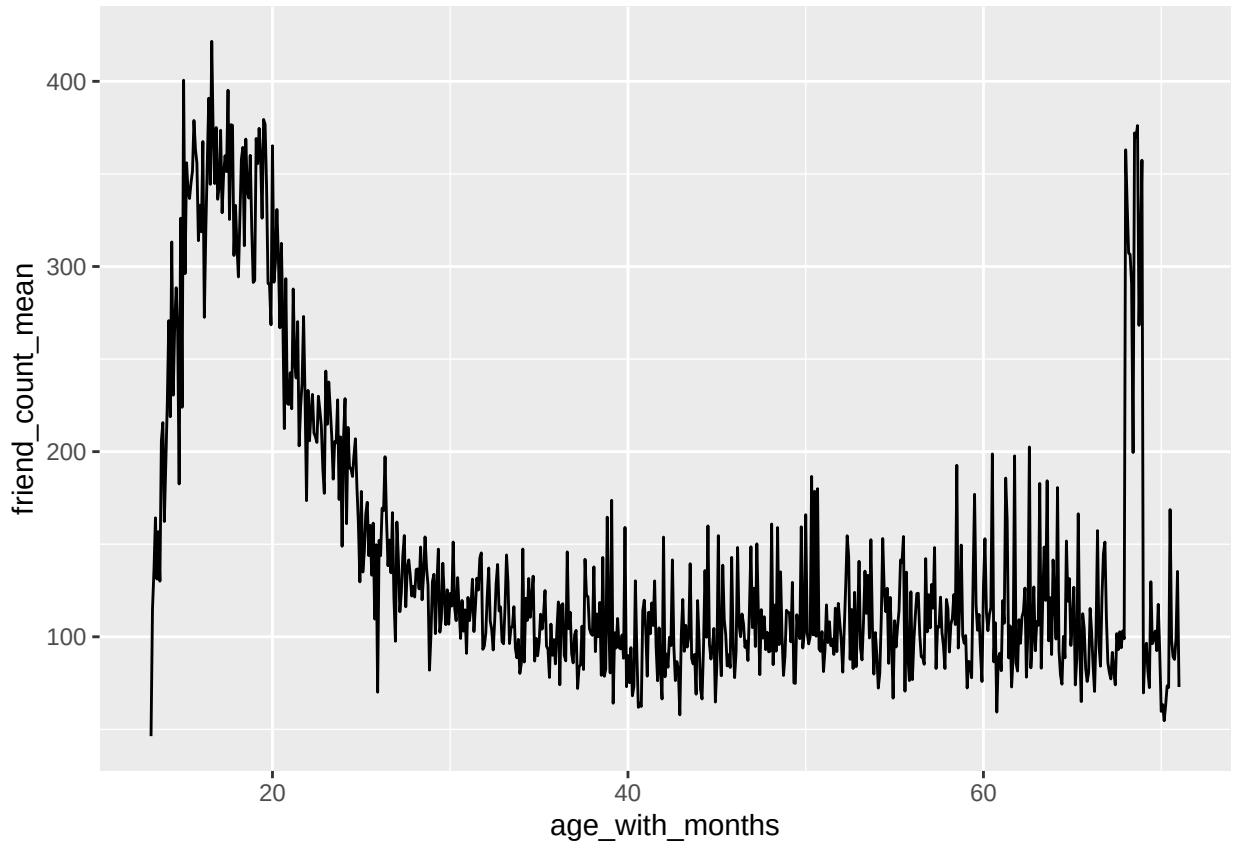
```
## # A tibble: 6 x 4
##   age_with_months friend_count_mean friend_count_median     n
##             <dbl>             <dbl>               <dbl> <int>
## 1            13.2              46.3                30.5     6
## 2            13.2             115.                 23.5    14
## 3            13.3             136.                 44      25
## 4            13.4             164.                 72      33
## 5            13.5             131.                 66      45
## 6            13.6             157.                 64      54
```

```
age_with_months_groups <- group_by(pf, age_with_months)
pf.fc_by_age_months2 <- summarise(age_with_months_groups,
                          friend_count_mean = mean(friend_count),
                          friend_count_median = median(friend_count),
                          n = n())
pf.fc_by_age_months2 <- arrange(pf.fc_by_age_months2, age_with_months)
head(pf.fc_by_age_months2)
```

```
## # A tibble: 6 x 4
##   age_with_months friend_count_mean friend_count_median     n
##             <dbl>             <dbl>               <dbl> <int>
## 1            13.2              46.3                30.5     6
## 2            13.2             115.                 23.5    14
## 3            13.3             136.                 44      25
## 4            13.4             164.                 72      33
## 5            13.5             131.                 66      45
## 6            13.6             157.                 64      54
```

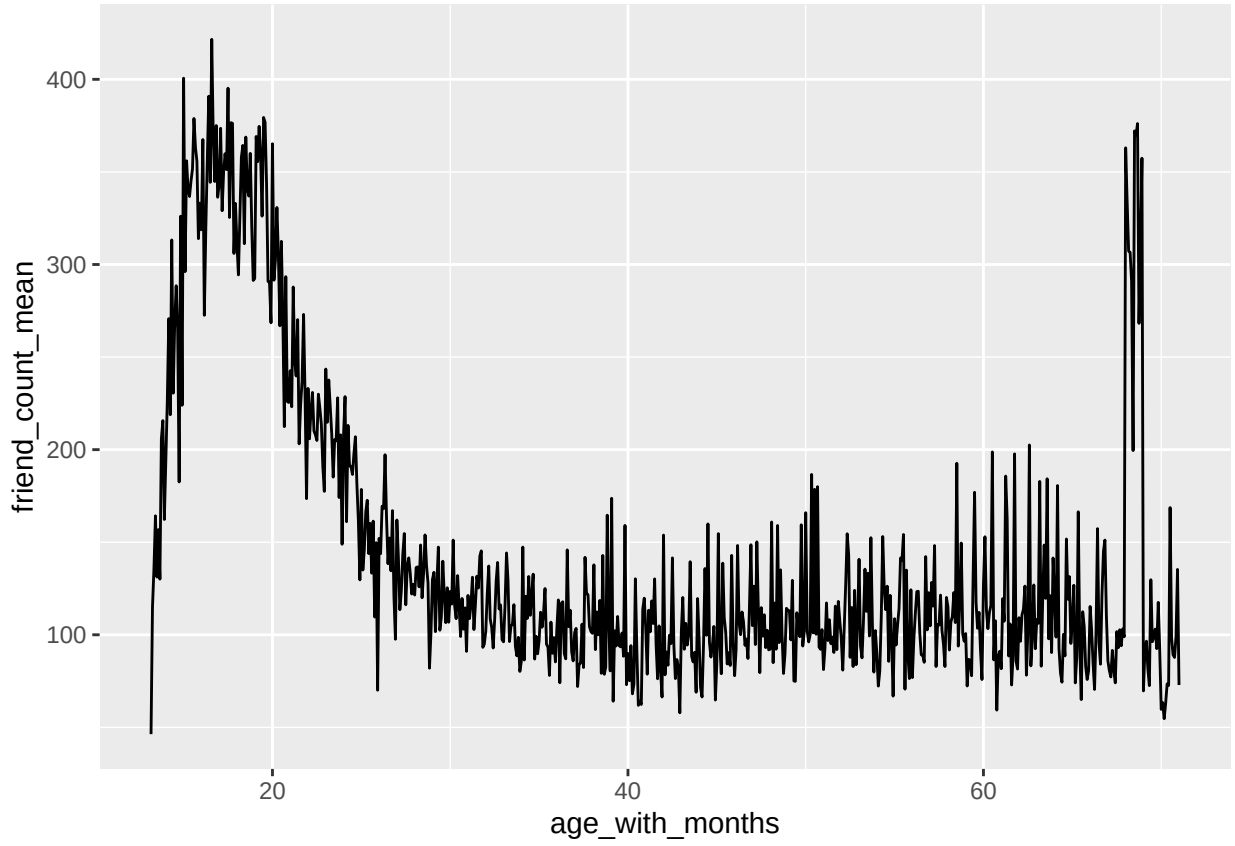
Programming Assignment

```
# Create a new line plot showing friend_count_mean versus the new variable,
# age_with_months. Be sure to use the correct data frame (the one you created
# in the last exercise) AND subset the data to investigate users with ages less
# than 71.

# ENTER YOUR CODE BELOW THIS LINE.
# ================================================================
ggplot(aes(x = age_with_months,
           y = friend_count_mean),
       data = subset(pf.fc_by_age_months, age_with_months <= 71)) +
  geom_line()
```



Noise in Conditional Means

```
ggplot(aes(x = age_with_months,
           y = friend_count_mean),
       data = subset(pf.fc_by_age_months, age_with_months <= 71)) +
  geom_line()
```



Smoothing Conditional Means

Notes:

```
p1 = ggplot(aes(x = age, y = friend_count_mean),
       data = subset(pf.fc_by_age, age < 71)) +
  geom_line() +
  geom_smooth()

p2 = ggplot(aes(x = age_with_months,
           y = friend_count_mean),
       data = subset(pf.fc_by_age_months, age_with_months <= 71)) +
  geom_line() +
  geom_smooth()

p3 = ggplot(aes(x = round(age / 5) * 5, y = friend_count),
       data = subset(pf, age < 71)) +
  stat_summary(fun.y = mean, geom = "line", color = "red", linetype = 2)
```

```
## Warning: `fun.y` is deprecated. Use `fun` instead.
```

```
library(gridExtra)
```

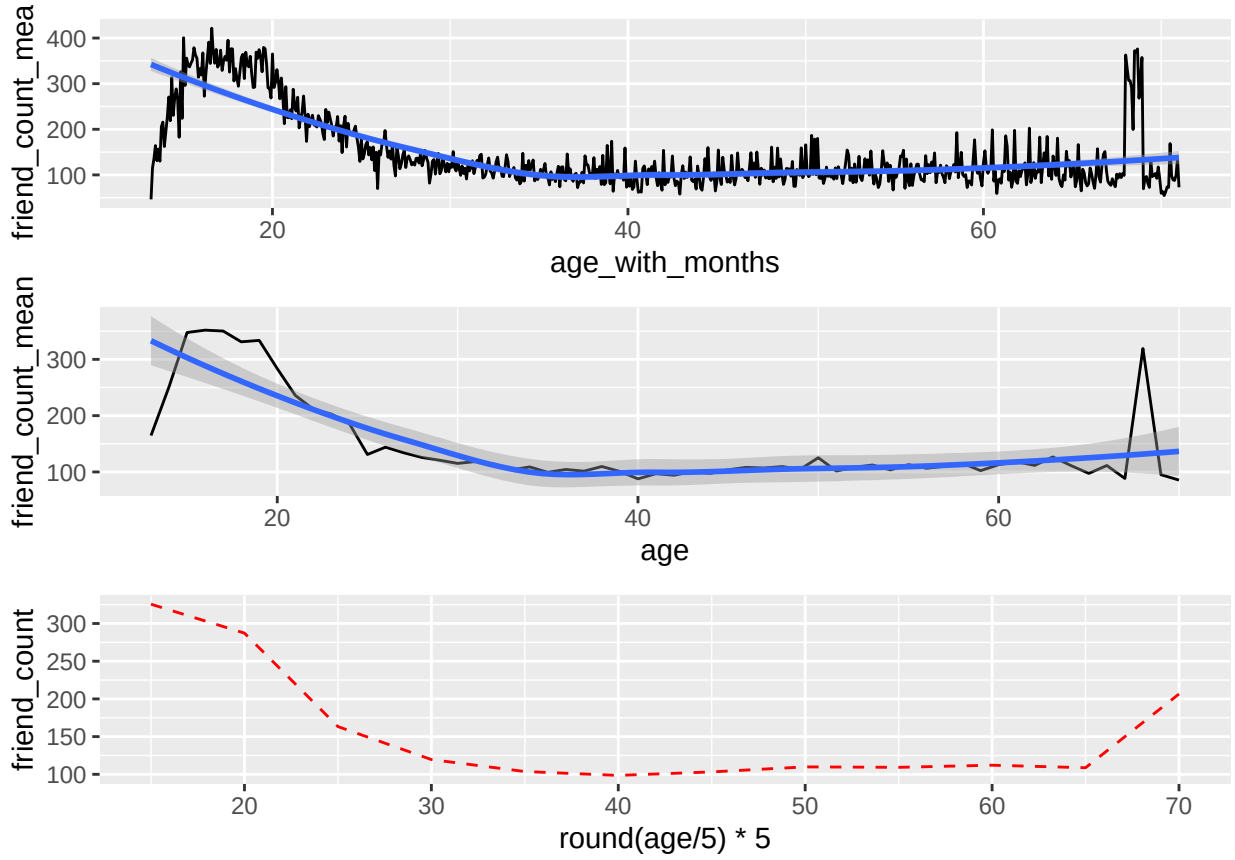
```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

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

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

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



Which Plot to Choose?

Notes:

Analyzing Two Variables

Reflection: 1. scatter 2. scatter with conditional limits 3.geom_line() 4.stat_summary() 5.correlation 6. geom_smooth()

Click **KnitHTML** to see all of your hard work and to have an html page of this lesson, your answers, and your notes!