Lesson 4

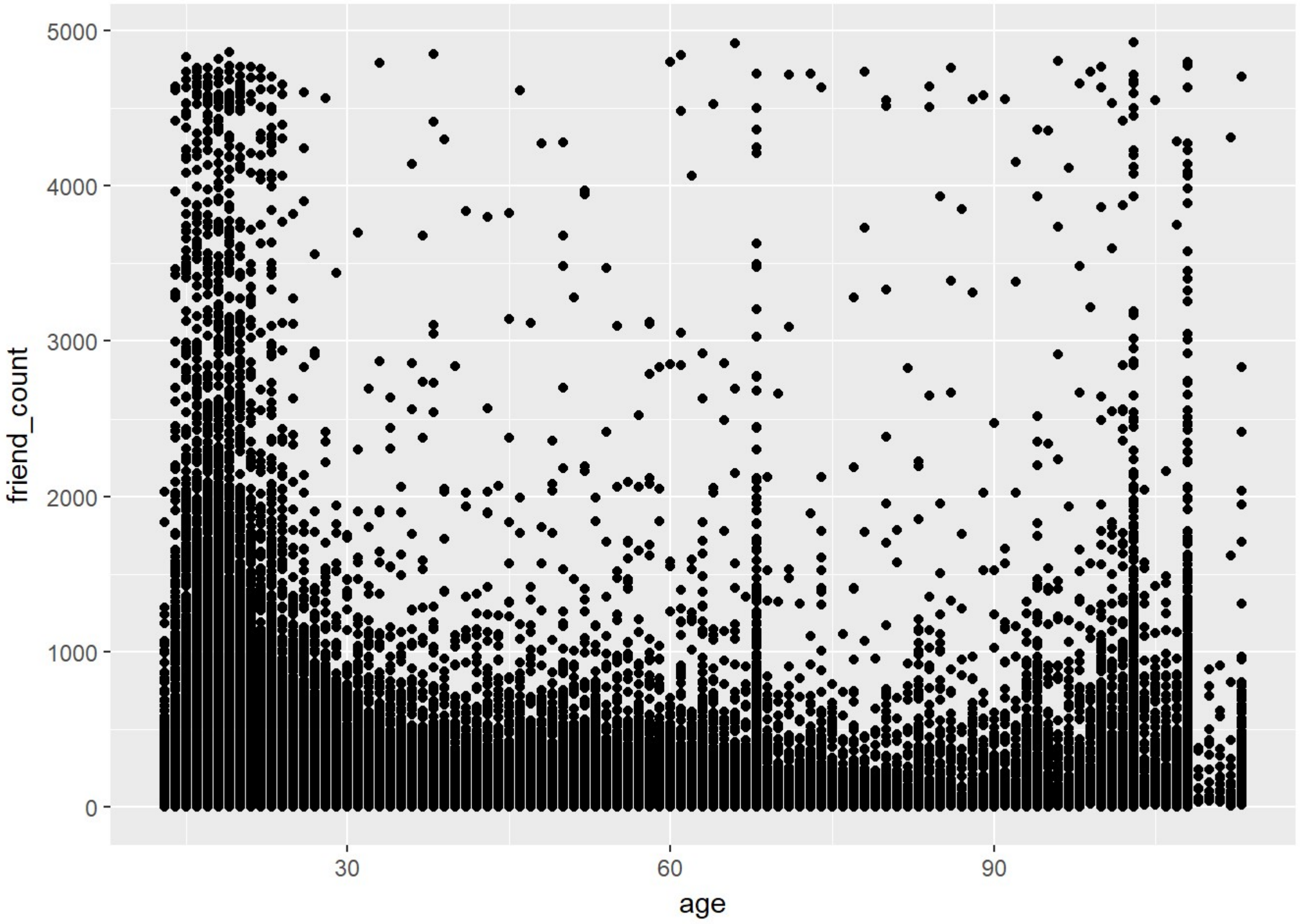
Scatterplots and Perceived Audience Size

Notes:

Scatterplots

Notes: qplot(age, friend_count, data = pf) can be used because qplot uses x, y format. ggplot(aes(x = age, y = friend_count), data = pf) + geom_point()

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

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



What are some things that you notice right away?

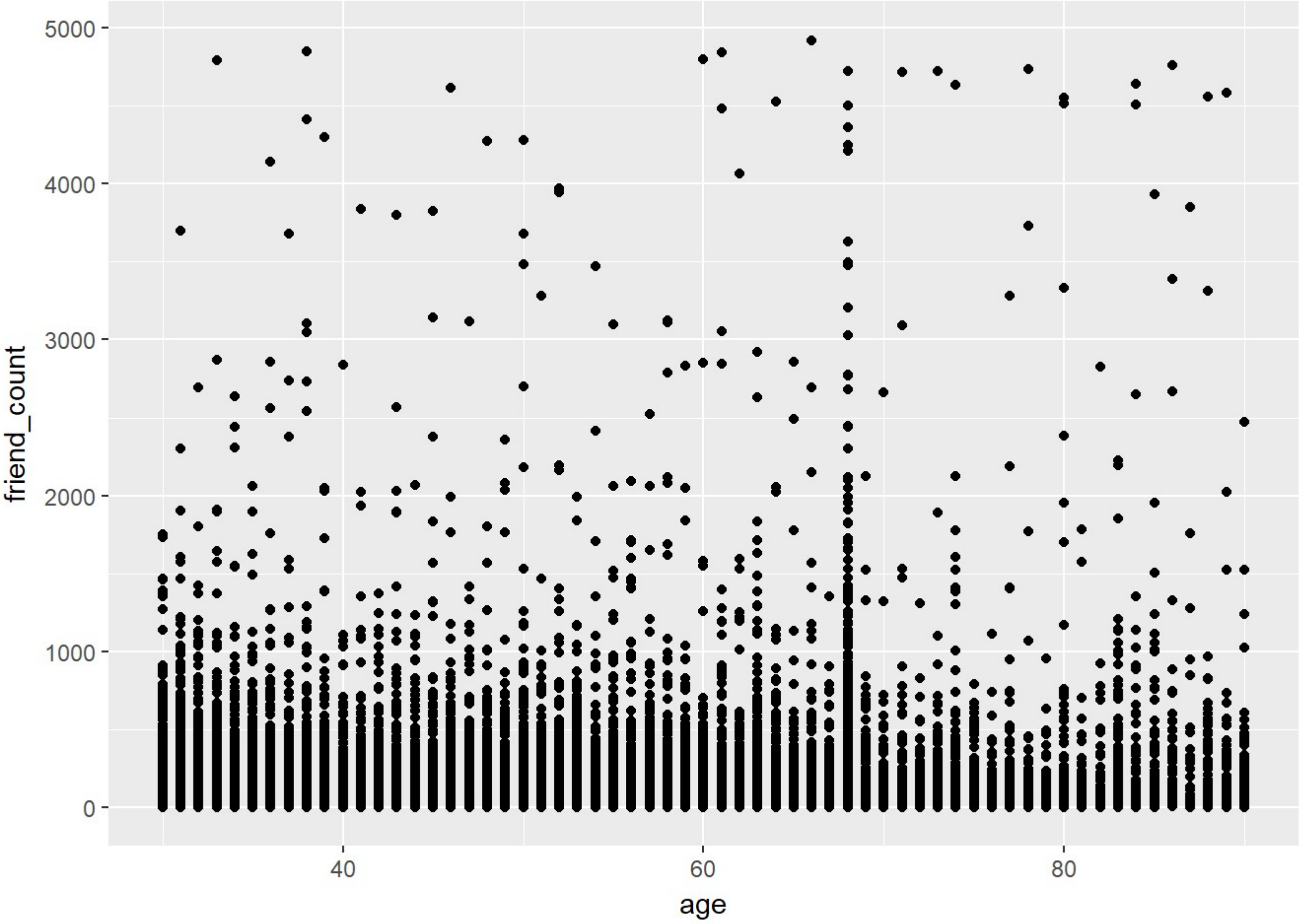
Response: ages extends beyond 90 and users lower than 30 have many more friends. ***

ggplot Syntax

Notes:

```
ggplot(aes(x = age, y = friend_count), data = pf) +
  geom_point() + xlim(30, 90)
```

```
## Warning: Removed 56588 rows containing missing values (geom_point).
```

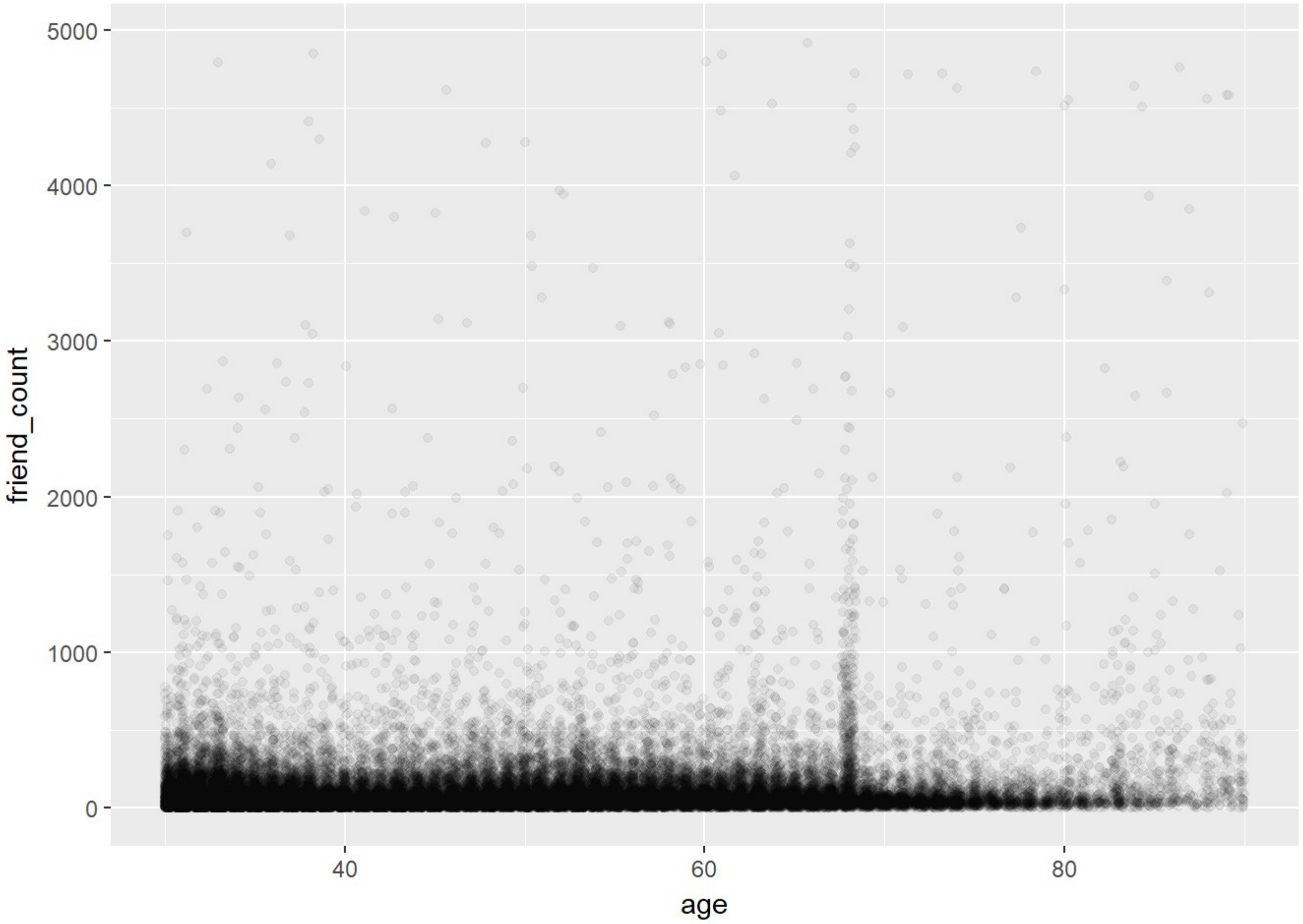


Overplotting

Notes: age is a continuous value. It is expressed as an int in the plot and as such makes the columns line up neatly which is not a natural state. By adding jitter we introduce noise to the data giving a more realistic reflection of the data dispersion.

```
ggplot(aes(x = age, y = friend_count), data = pf) +
  geom_jitter(alpha = 1/20) +
  xlim(30, 90)
```

```
## Warning: Removed 57476 rows containing missing values (geom_point).
```



What do you notice in the plot?

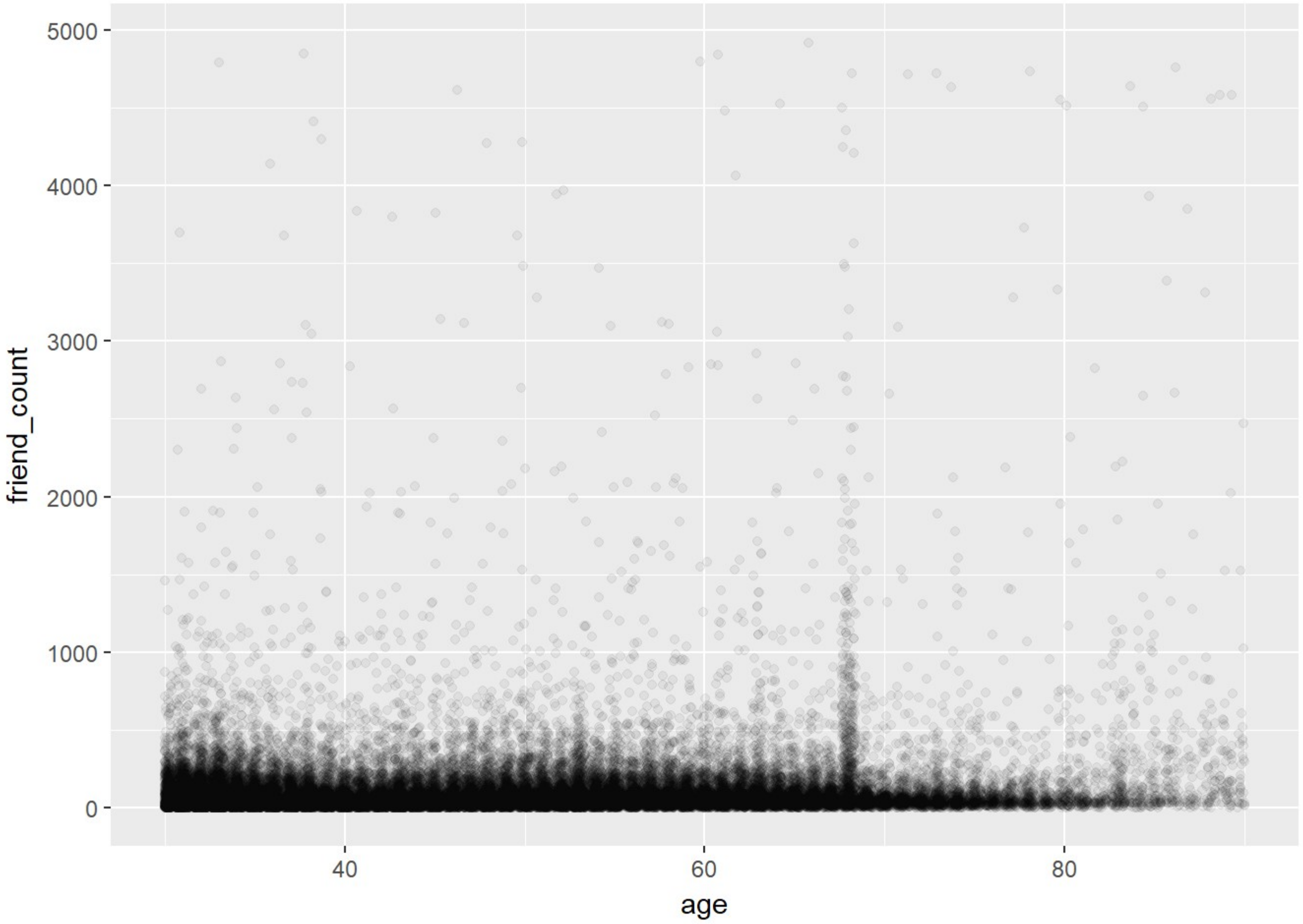
Response: The information scatter feels more realistic and a truer reflection of how the data would be represented. It is also clearer that the age value of 69 is inaccurate. Younger users do not seem to be as high as they were before. ***

Coord_trans()

Notes:

```
ggplot(aes(x = age, y = friend_count), data = pf) +
  geom_jitter(alpha = 1/20) +
  xlim(30, 90)
```

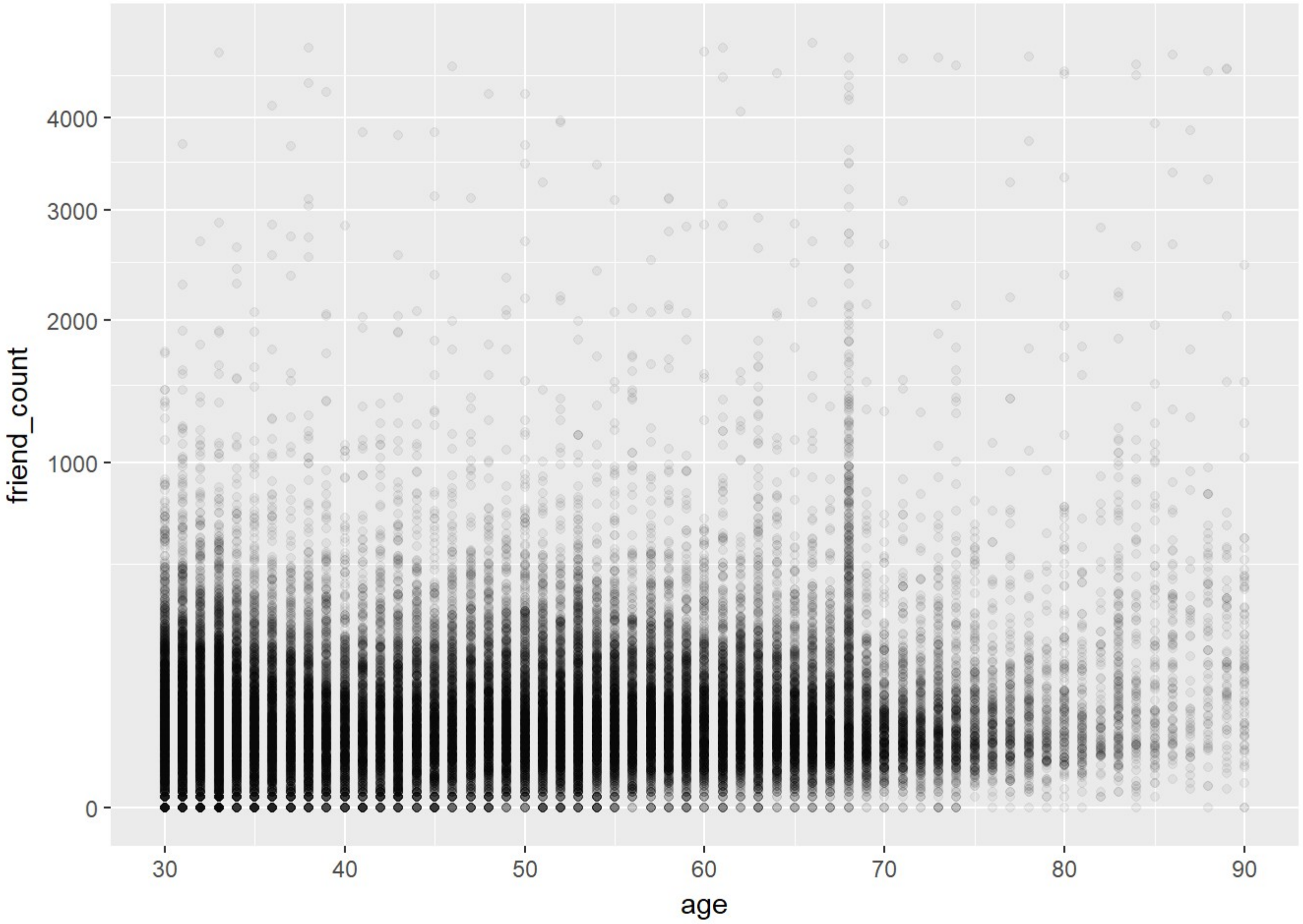
```
## Warning: Removed 57476 rows containing missing values (geom_point).
```



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) +
  xlim(30, 90) +
  coord_trans(y = 'sqrt')
```

```
## Warning: Removed 56588 rows containing missing values (geom_point).
```



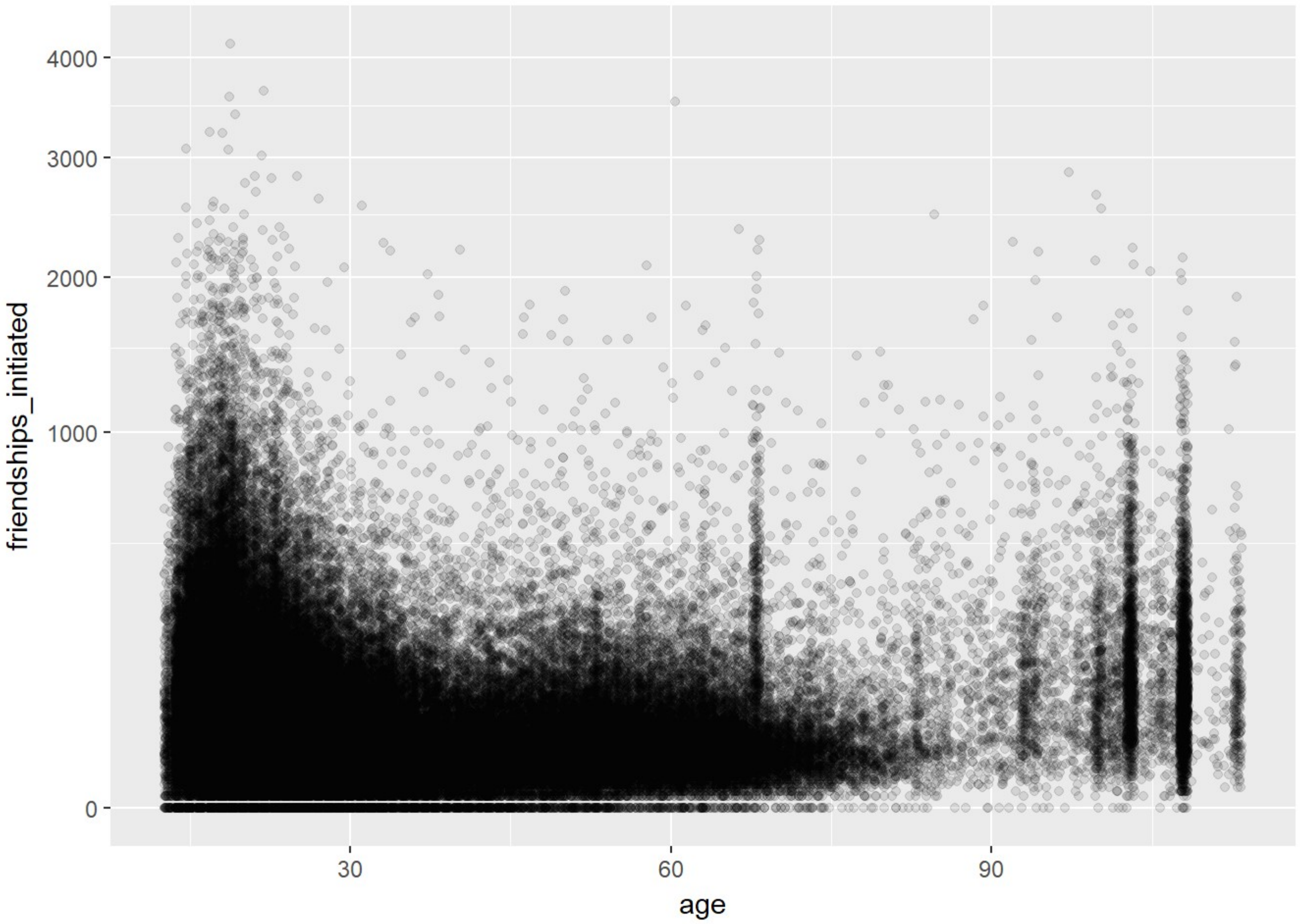
What do you notice?

By adding the sqrt transform it zooms in on the datapoints and we can see that actual count concentration is well below 1000. ***

Alpha and Jitter

Notes:

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



Overplotting and Domain Knowledge

Notes:

Conditional Means

Notes:

```
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
```

```
pf %>%
  group_by(age) %>%
  summarise(friend_count_mean = mean(friend_count),
            friend_count_median = median(friend_count),
            n = n()) %>%
  arrange(age)
```

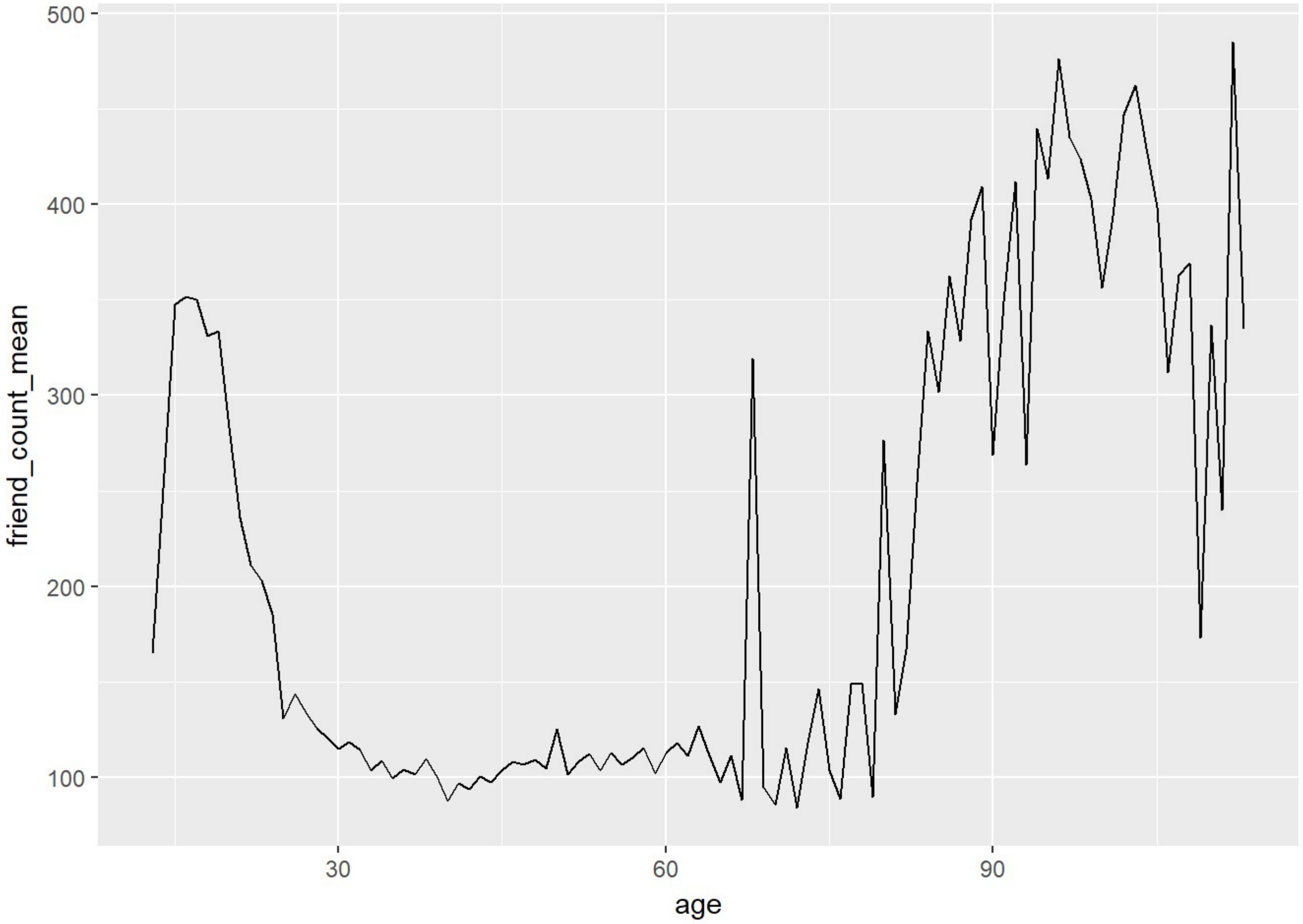
```
## # A tibble: 101 x 4
##      age friend_count_mean friend_count_median     n
##    <int>             <dbl>              <dbl>  <int>
##  1    13              165.                  74   484
##  2    14              251.                 132  1925
##  3    15              348.                 161  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
## # ... with 91 more rows
```

```
head(pf.fc_by_age, 20)
```

```
## # A tibble: 20 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
## 16    28             126.               66    2364
## 17    29             121.               66    1936
## 18    30             115.               67.5  1716
## 19    31             118.               63    1694
## 20    32             114.               63    1443
```

Create your plot!

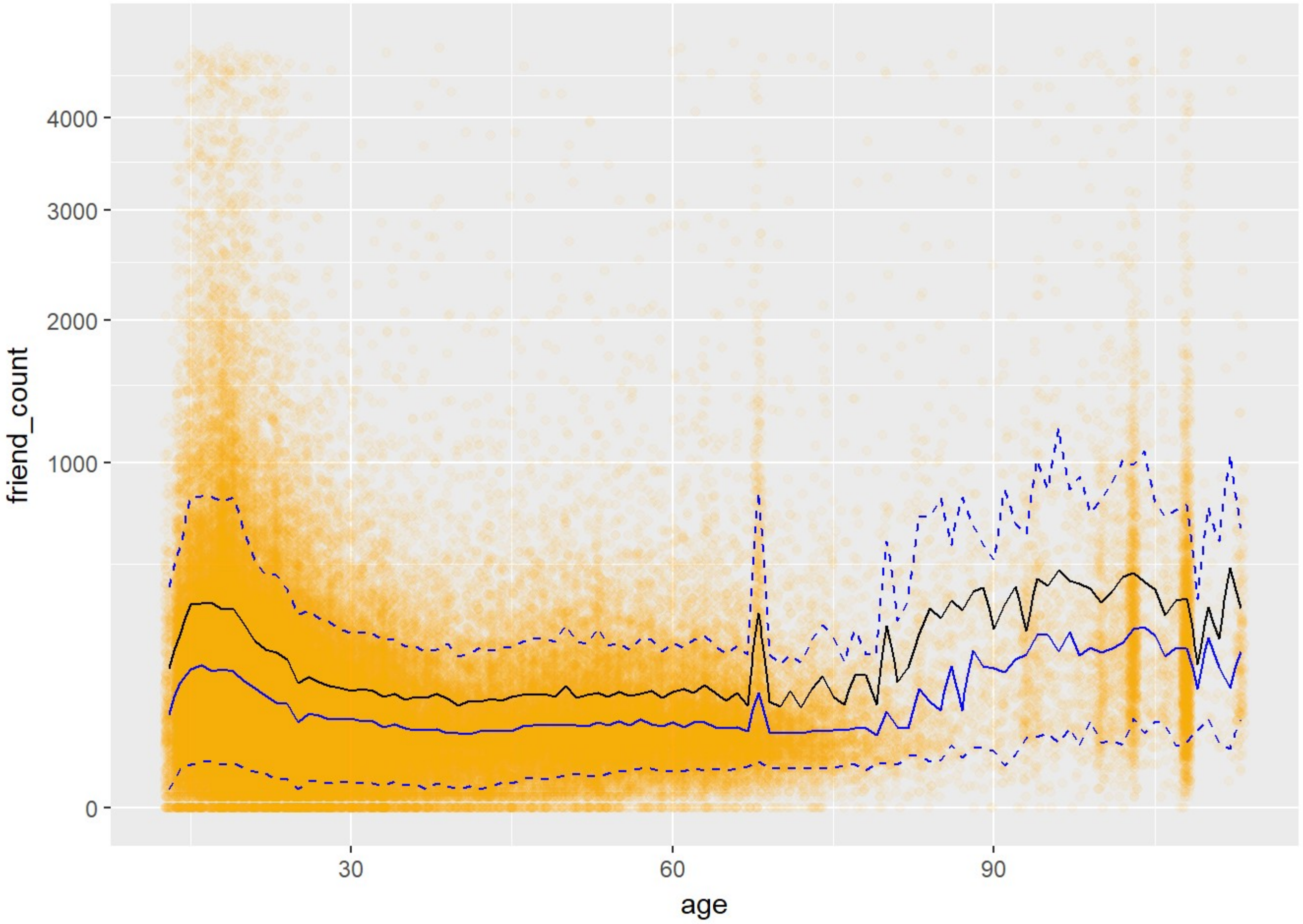
```
ggplot(aes(age, friend_count_mean),data = pf.fc_by_age) +
  geom_line()
```



Overlaying Summaries with Raw Data

Notes: fun.y take a function and applies it

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



What are some of your observations of the plot?

Response:

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:

```
cor.test(pf$age, pf$friend_count, method = 'pearson')
```

```
##
##  Pearson's product-moment correlation
##
## data:  pf$age and pf$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
```

```
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: -0.02740737

Correlation on Subsets

Notes:

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

```
##
##  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
```

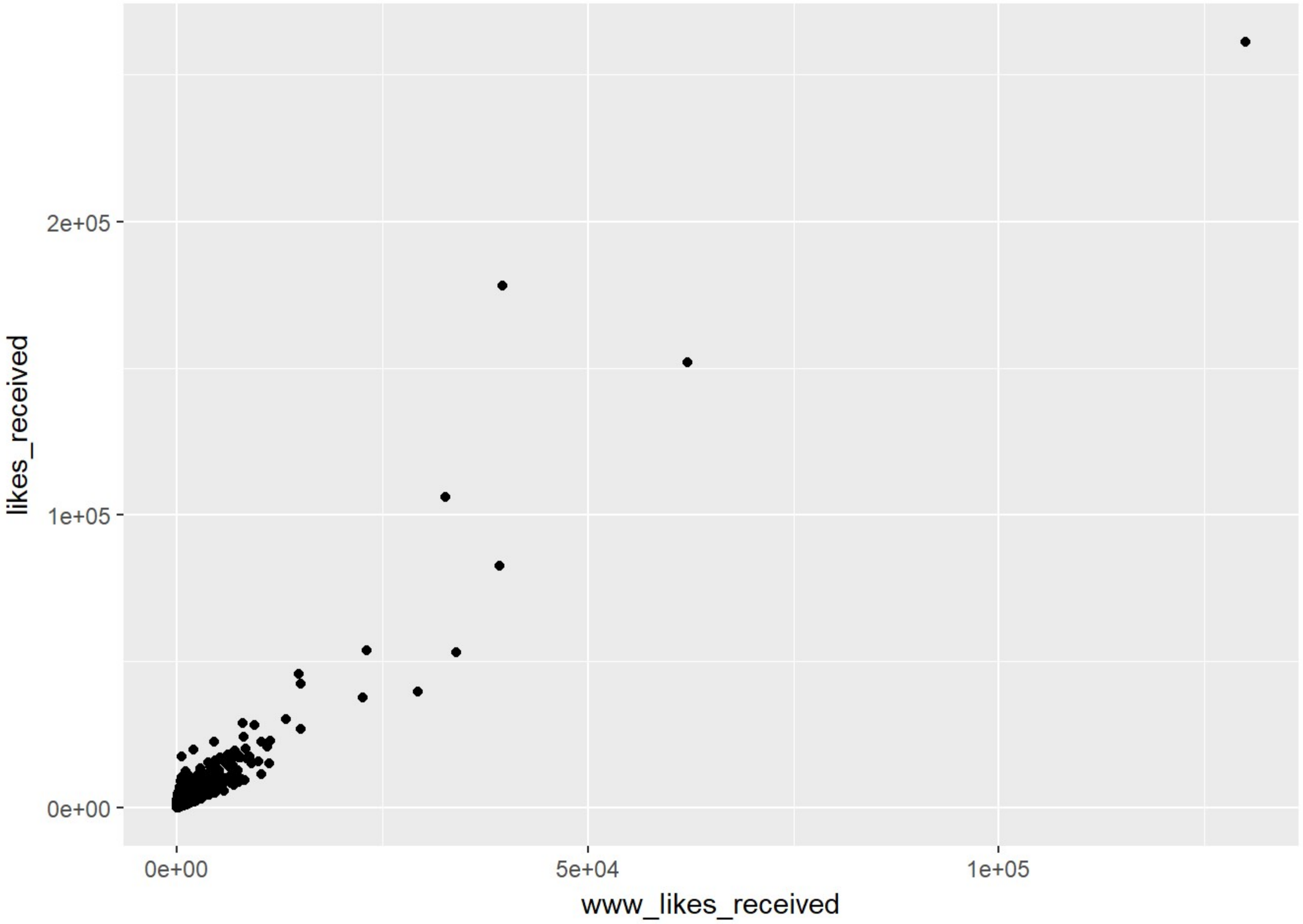
Correlation Methods

Notes:

Create Scatterplots

Notes:

```
ggplot(aes(www_likes_received, y = likes_received), data = pf) +
  geom_point()
```



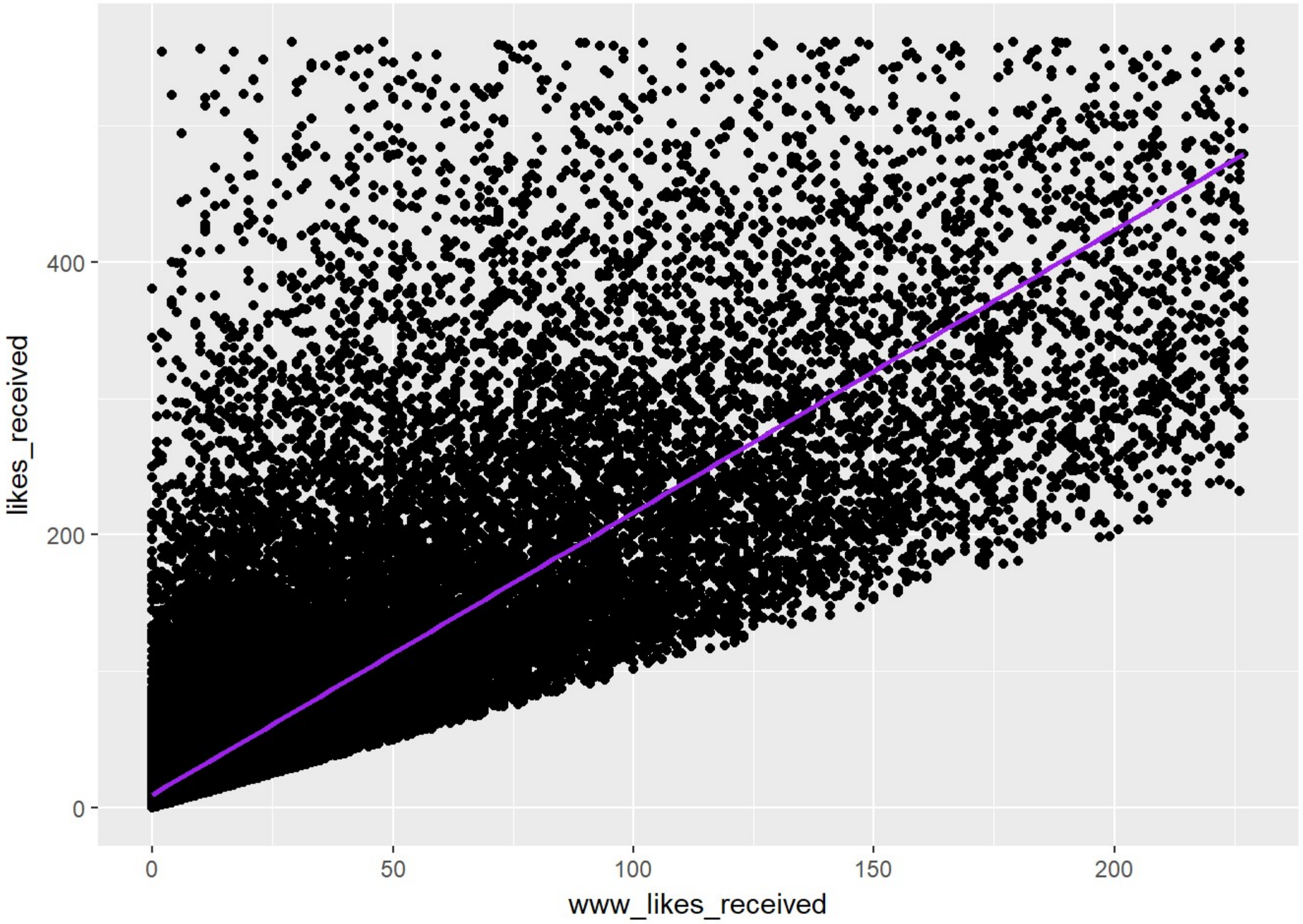
Strong Correlations

Notes:

```
ggplot(aes(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 = 'purple')
```

```
## 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, method = 'pearson'))
```

```
##
##  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:

```
library(alr3)
```

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

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

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

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

```
data("Mitchell")
?Mitchell
```

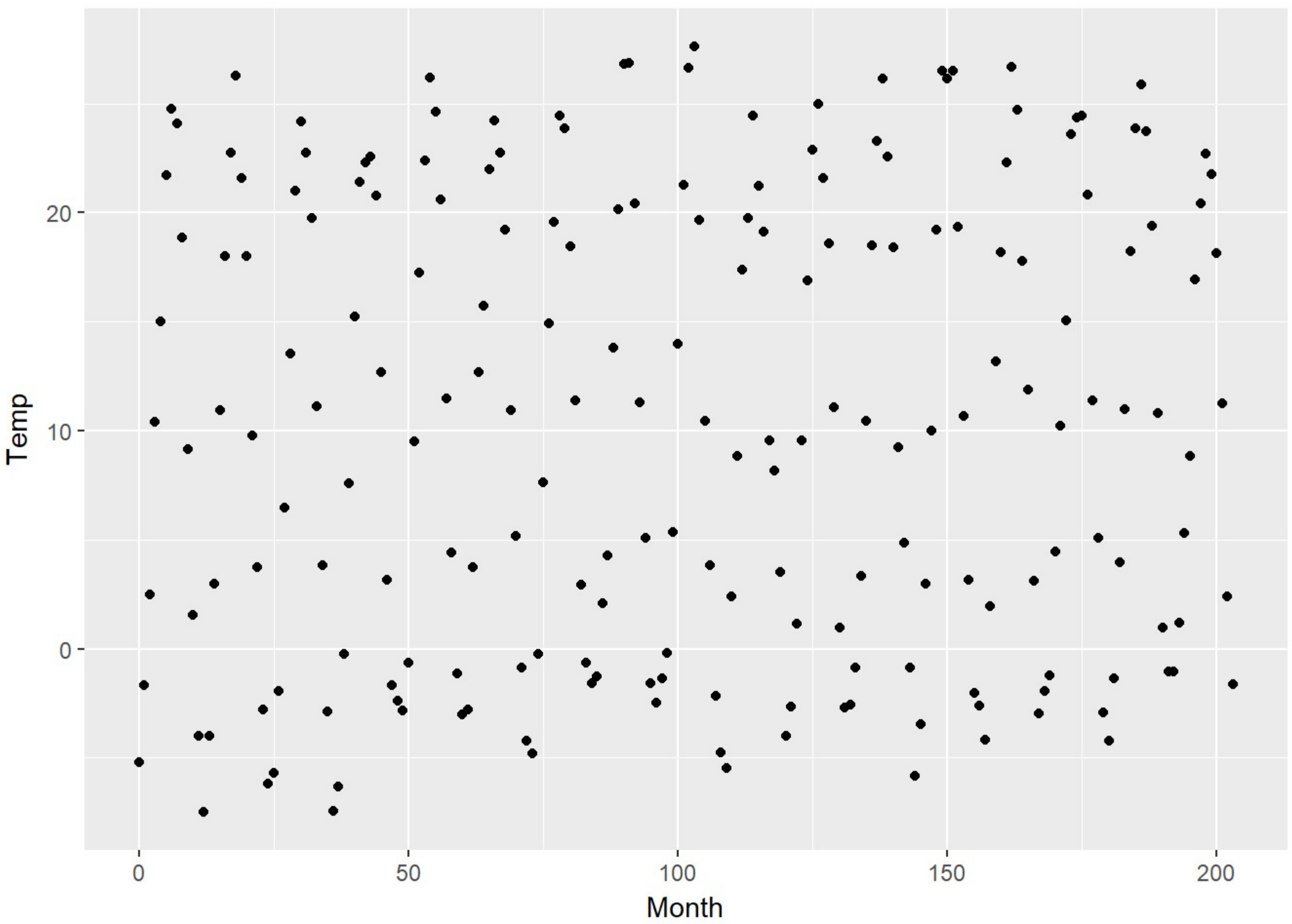
```
## starting httpd help server ...
```

```
##  done
```

```
str(Mitchell)
```

```
## 'data.frame':    204 obs. of  2 variables:
##  $ Month: int  0 1 2 3 4 5 6 7 8 9 ...
##  $ Temp : num  -5.18 -1.65 2.49 10.4 14.99 ...
```

```
ggplot(aes(x = Month, y = Temp), data = Mitchell) +
  geom_point()
```



Create your plot!

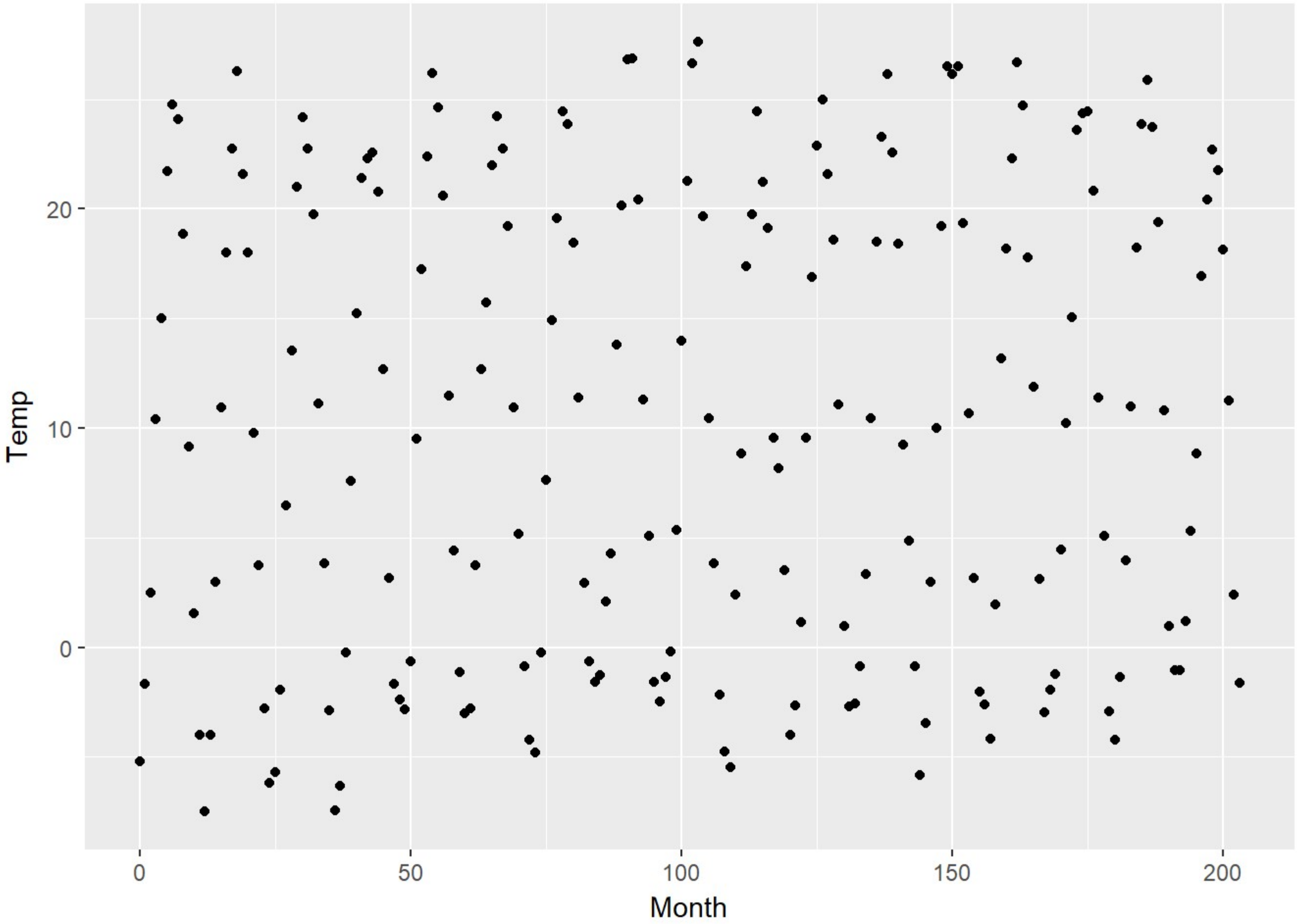
```
range(Mitchell$Month)
```

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

```
cor.test(Mitchell$Month, Mitchell$Temp, method = 'pearson')
```

```
##
##  Pearson's product-moment correlation
##
## data:  Mitchell$Month and Mitchell$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
```

```
ggplot(aes(x = Month, y = Temp), data = Mitchell) +
  geom_point()
```



Noisy Scatterplots

a. Take a guess for the correlation coefficient for the scatterplot.

b. What is the actual correlation of the two variables? (Round to the thousandths place)

Making Sense of Data

Notes:

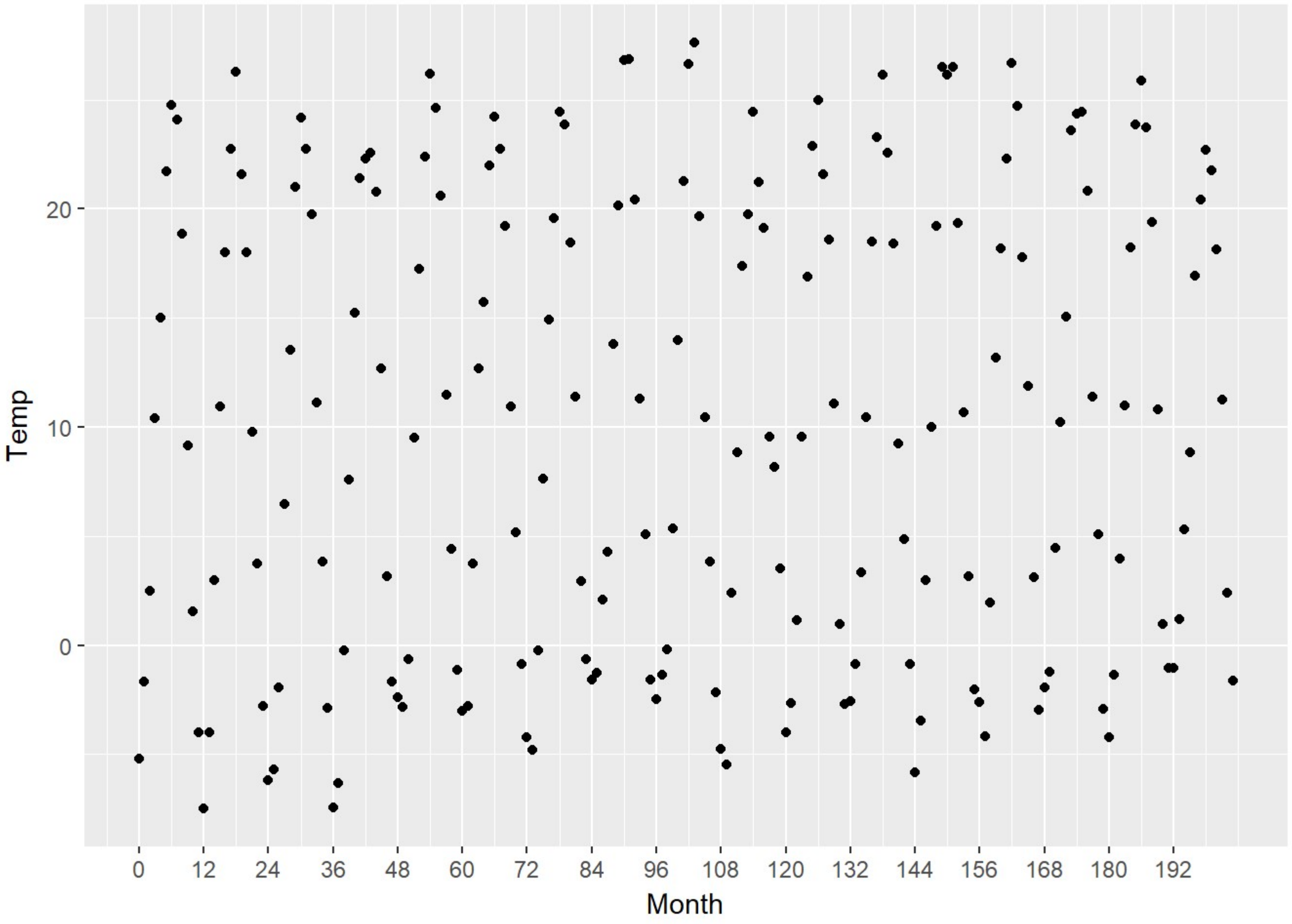
```
range(Mitchell$Month)
```

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

```
cor.test(Mitchell$Month, Mitchell$Temp, method = 'pearson')
```

```
##
##  Pearson's product-moment correlation
##
## data:  Mitchell$Month and Mitchell$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
```

```
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:

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

Understanding Noise: Age to Age Months

Notes:

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

Age with Months Means

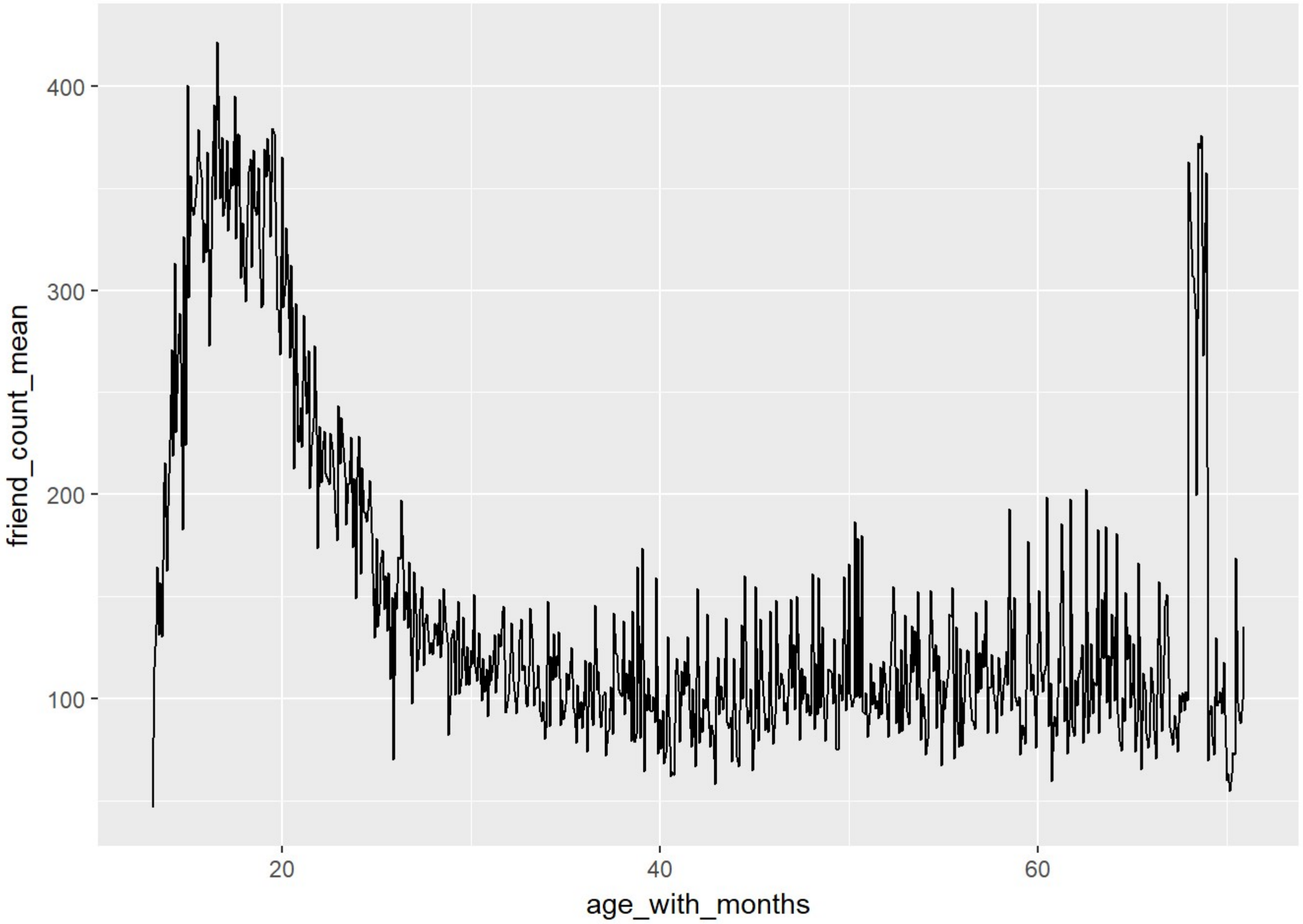
```
pf.fc_by_age_months <- pf %>%
  group_by(age_with_months) %>%
  summarise(friend_count_mean = mean(friend_count),
            friend_count_median = median(friend_count),
            n = n())%>%
  arrange(age_with_months)
```

Programming Assignment

```
library(dplyr)
pf.age_with_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)
```

Noise in Conditional Means

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



Smoothing Conditional Means

Notes: Bias variant tradeoff.

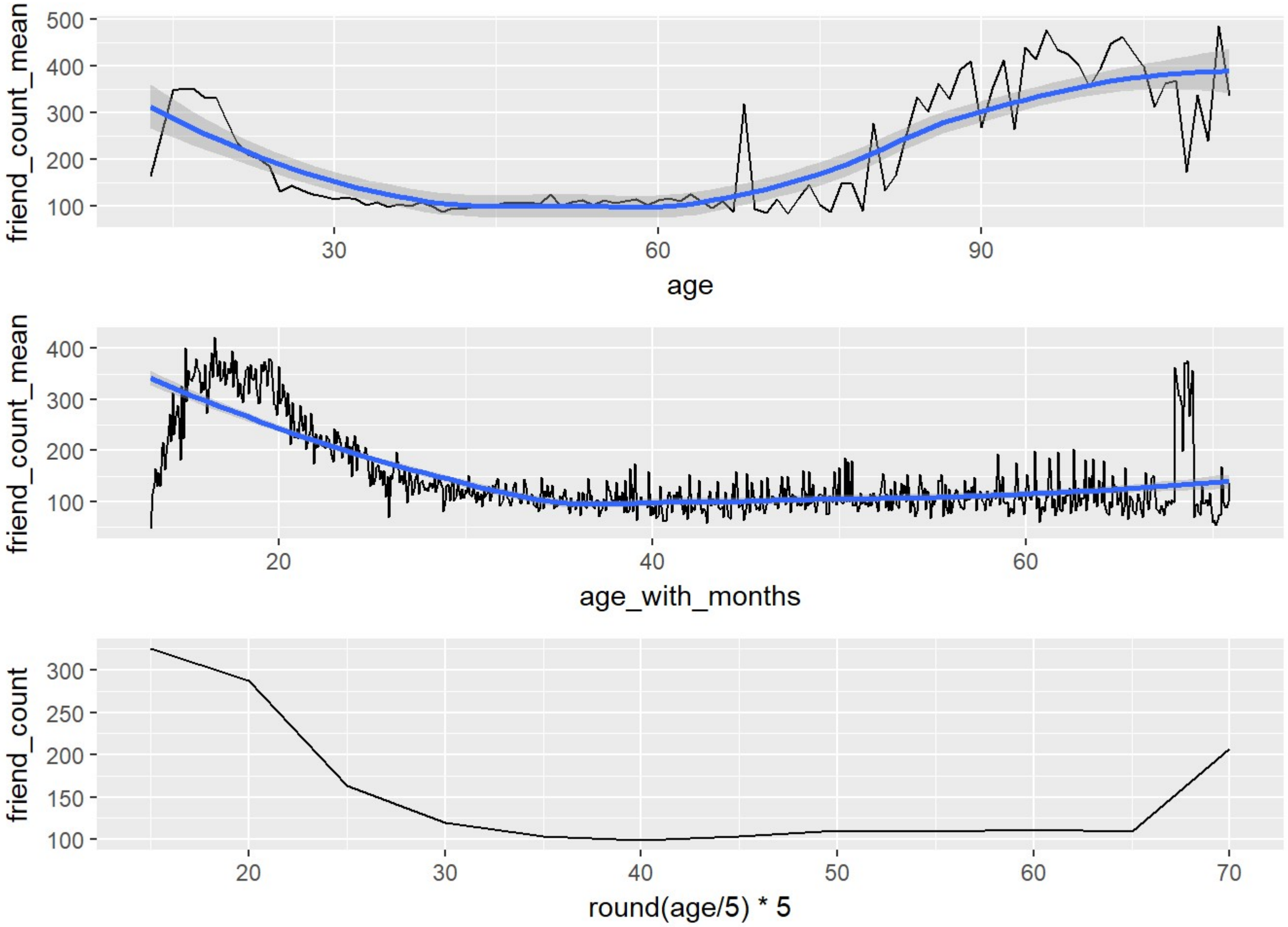
```
p1 <- ggplot(aes(x = age, y = friend_count_mean),
       data = pf.fc_by_age) +
  geom_line()+
  geom_smooth()

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)) +
  geom_line(stat = 'summary', fun.y = mean)
library(gridExtra)
```

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

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

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

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



Which Plot to Choose?

Notes: You don’t have to choose. We will often create multiple plots and summaries of the data. They can reveal different things about the data. When publishing you may want to narrow the scope to the plots that best communicate the findings. ***

Analyzing Two Variables

Reflection:

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