Lesson 5

Multivariate Data

Moira Perceived Audience Size Colored by Age

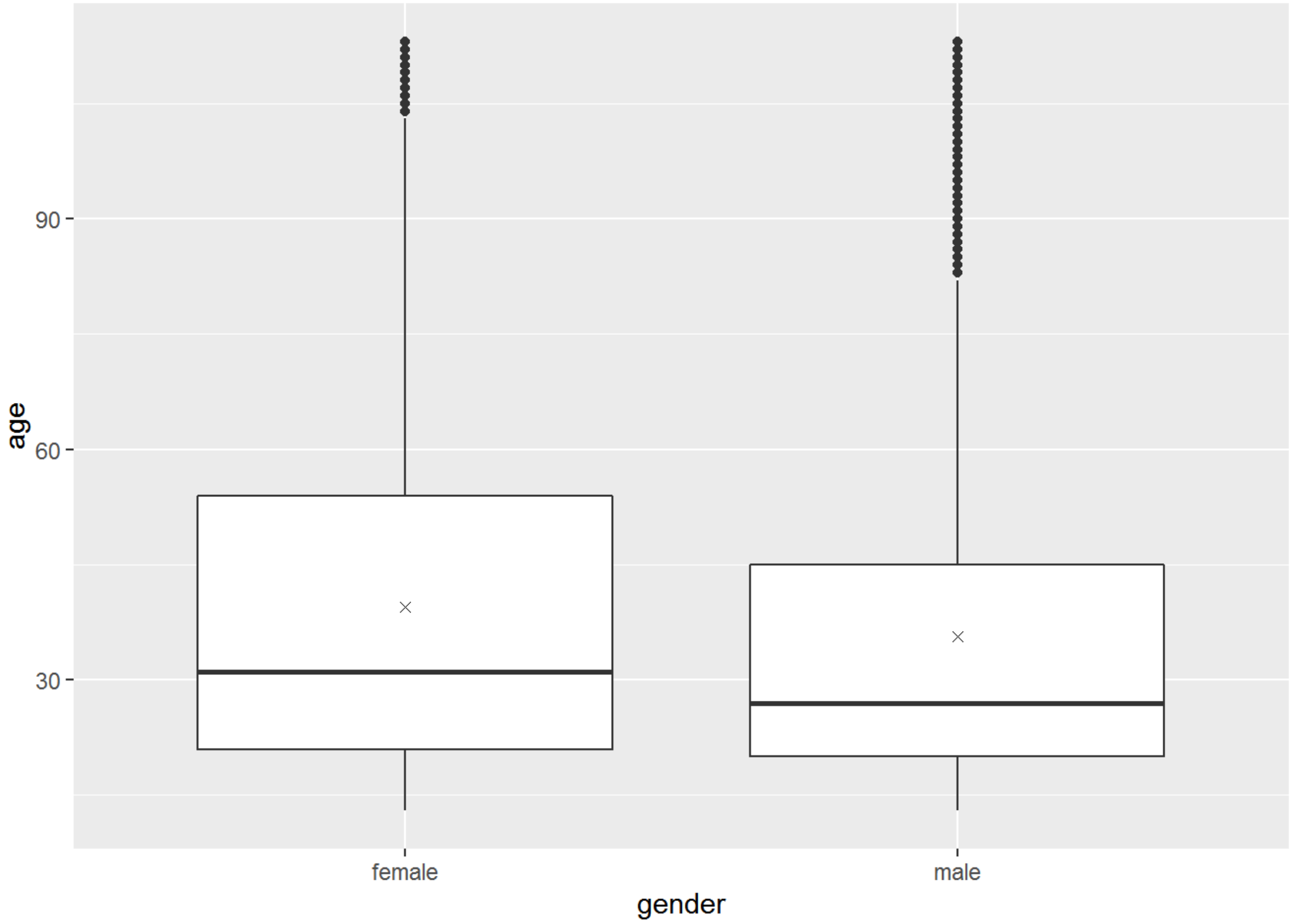
Third Qualitative Variable

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

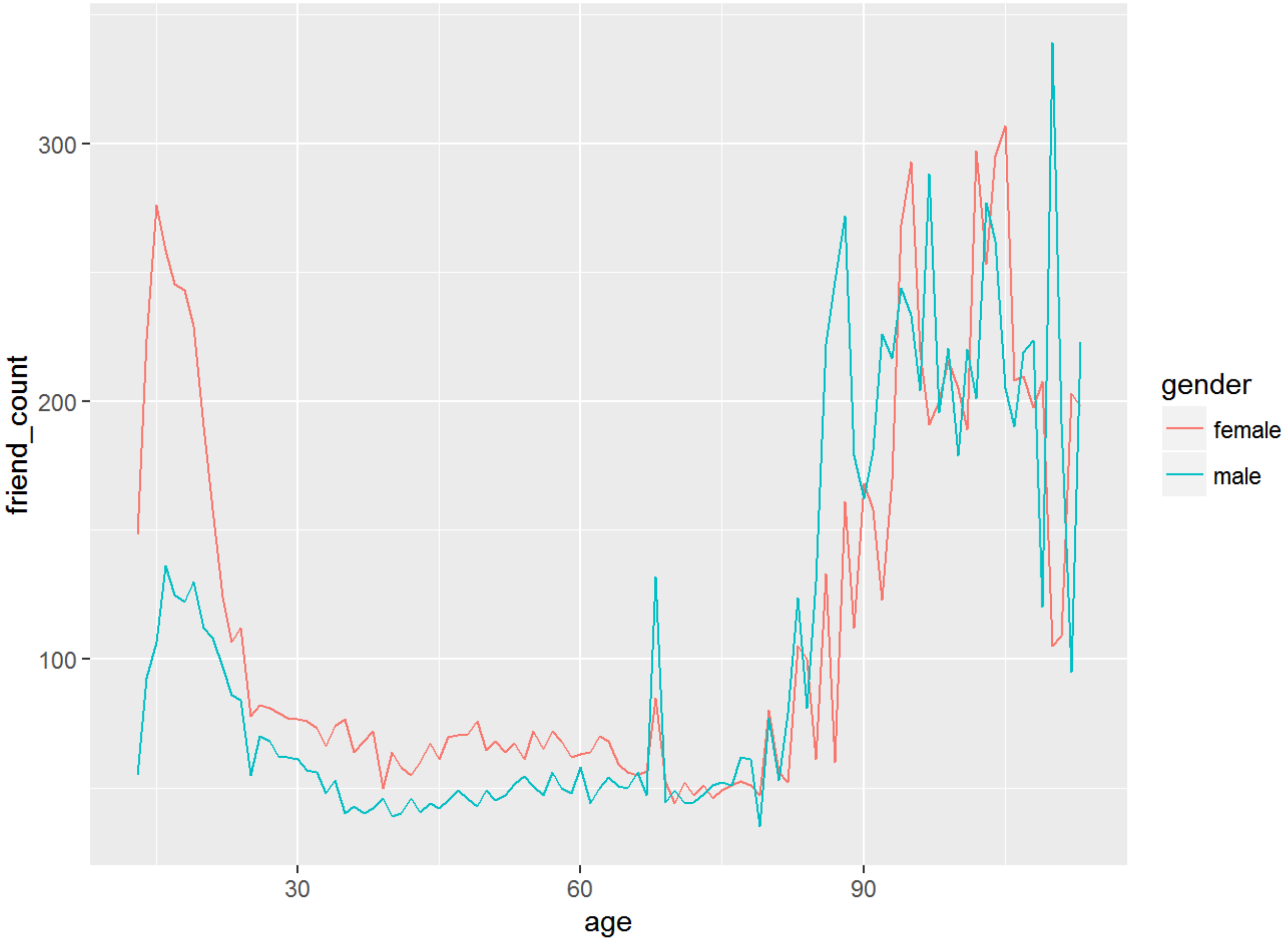
```
##  [1] "userid"                "age"
##  [3] "dob_day"               "dob_year"
##  [5] "dob_month"             "gender"
##  [7] "tenure"                "friend_count"
##  [9] "friendships_initiated" "likes"
## [11] "likes_received"        "mobile_likes"
## [13] "mobile_likes_received" "www_likes"
## [15] "www_likes_received"
```

```
library(ggplot2)
```

```
# ggplot(aes(x = gender, y = age),
# data = subset(pf, !is.na(gender))) + geom_histogram()
library(ggplot2)
ggplot(aes(x = gender, y = age),
       data = subset(pf, !is.na(gender))) +
  geom_boxplot() +
  stat_summary(fun.y = mean, geom = "point", shape = 4)
```



```
ggplot(aes(x = age, y = friend_count),
       data = subset(pf, !is.na(gender))) +
  geom_line(aes(color = gender) ,stat = "summary", fun.y = median)
```



```
# run this first !
library(dplyr)
```

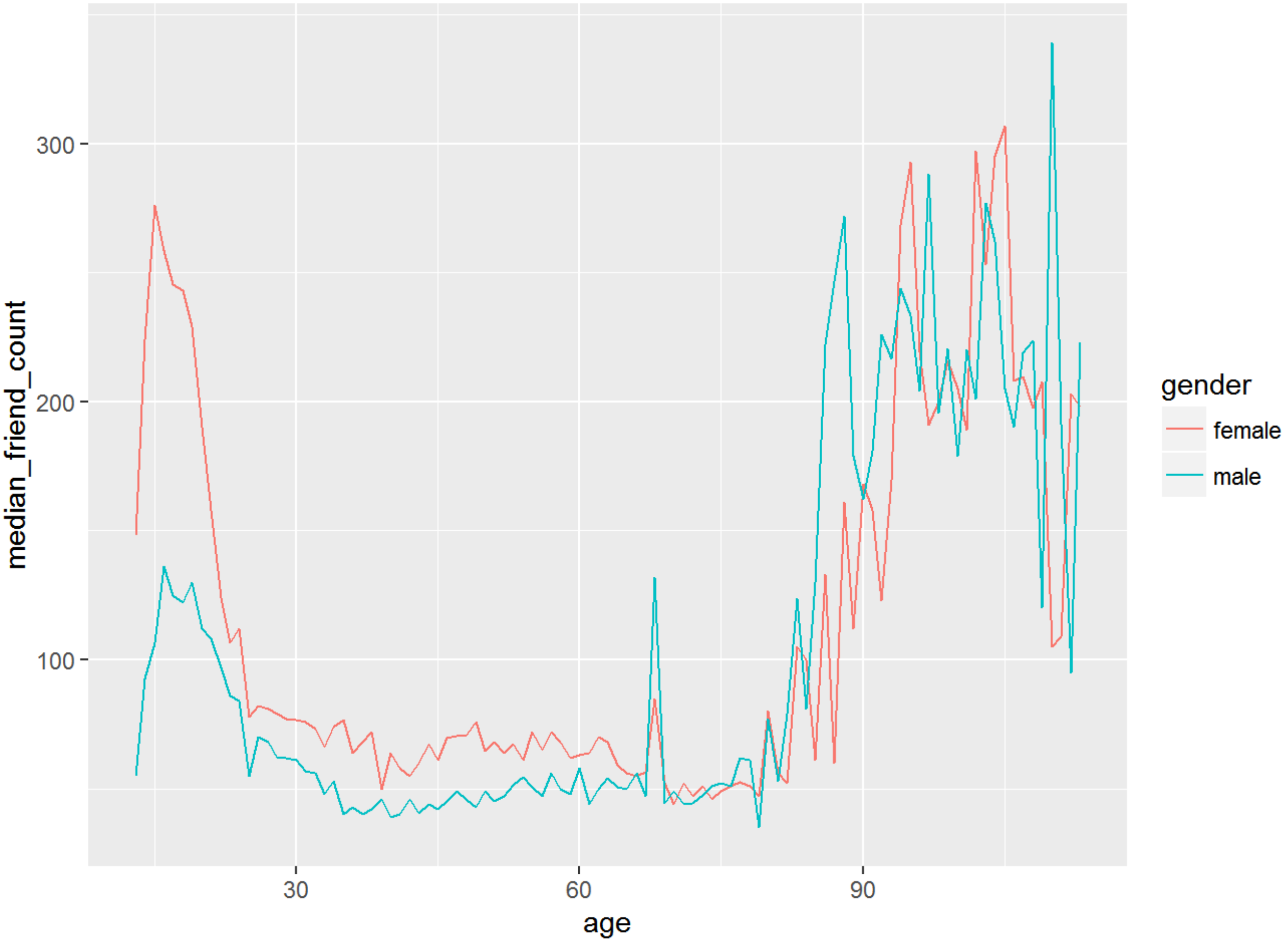
```
##
## Attaching package: 'dplyr'
##
## The following objects are masked from 'package:stats':
##
##     filter, lag
##
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
# method 1: no pipe
# no subset data with na data (gender), 274 observations
pf.fc_by_age_gender <- select(pf, age, gender, friend_count)
pf.fc_by_age_gender <- group_by(pf, age, gender)
pf.fc_by_age_gender <- pf.fc_by_age_gender %>% summarise(mean_friend_count = mean(frien
d_count),
                                                     median_friend_count = median(frien
d_count),
                                                     n = n())

# alternate method: pipe & subset na (gender), 202 observations
# the last ungroup(): grouping by 2 var collapses last
#
pf.fc_by_age_gender <- pf %>%
  filter(!is.na(gender)) %>%
  group_by(age, gender) %>%
  summarise(mean_friend_count = mean(friend_count),
            median_friend_count = median(friend_count),
            n = n()) %>%
  ungroup() %>%
  arrange(age)
```

Plotting Conditional Summaries

```
ggplot(aes(x = age, y = median_friend_count),
       data = subset(pf.fc_by_age_gender, !is.na(gender))) +
  geom_line(aes(color = gender) ,stat = "summary", fun.y = median)
```



Thinking in Ratios

Wide and Long Format

Reshaping Data

Notes:

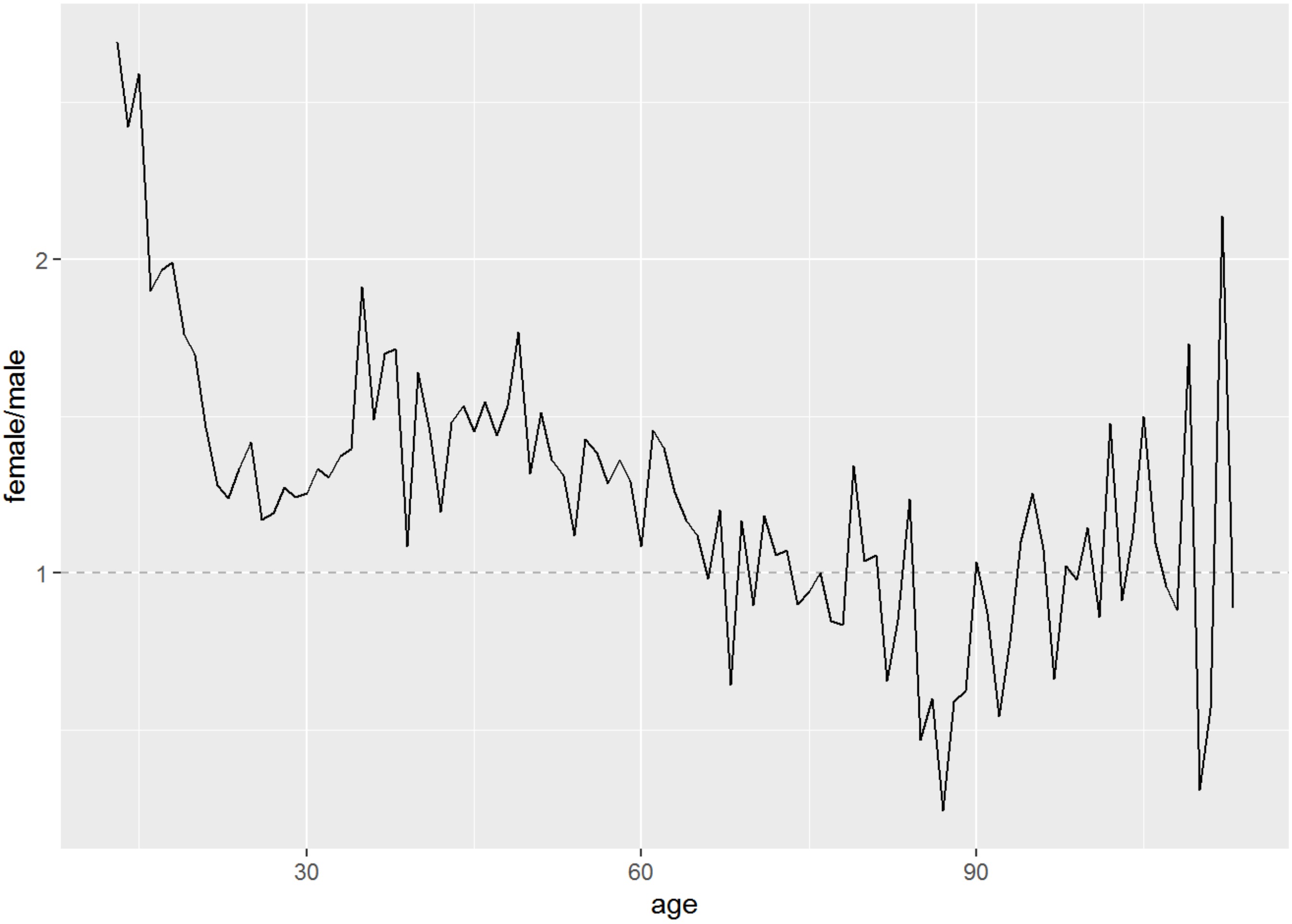
```
# It is possible to do it with tidyr and dplyr:
#pf.fc_by_age_gender.wide <-     subset(pf.fc_by_age_gender[c('age', 'gender', 'median_
friend_count')],                #   !is.na(gender)) %>%
#    spread(gender, median_friend_count) %>%
#    mutate(ratio = male / female)

# using now reshape2
#install.packages('reshape2')
library(reshape2)

pf.fc_by_age_gender.wide <- dcast(pf.fc_by_age_gender,
                                  age ~ gender,
                                  value.var = "median_friend_count")
```

Ratio Plot

```
library(ggplot2)
ggplot(aes(x = age, y = female / male),
       data = subset(pf.fc_by_age_gender.wide, !is.na(age))) +
  geom_line() +
  geom_hline(aes(yintercept = 1), alpha = 0.3, linetype = "dashed")
```



Third Quantitative Variable

```
# Create variable year_joined and assign to pf dataframe
# tenure is in days and we want years so /365
# floor function
pf$year_joined <- floor(2014 - pf$tenure/365)
```

Cut a Variable

```
summary(pf$year_joined)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##    2005    2012    2012    2012    2013    2014       2
```

```
table(pf$year_joined)
```

```
##
##  2005  2006  2007  2008  2009  2010  2011  2012  2013  2014
##     9    15   581  1507  4557  5448  9860 33366 43588    70
```

```
#        (2004, 2009]
#        (2009, 2011]
#        (2011, 2012]
#        (2012, 2014]
pf$year_joined.bucket <- cut(pf$year_joined,
                             c(2004, 2009, 2011, 2012, 2014))
```

Plotting it All Together

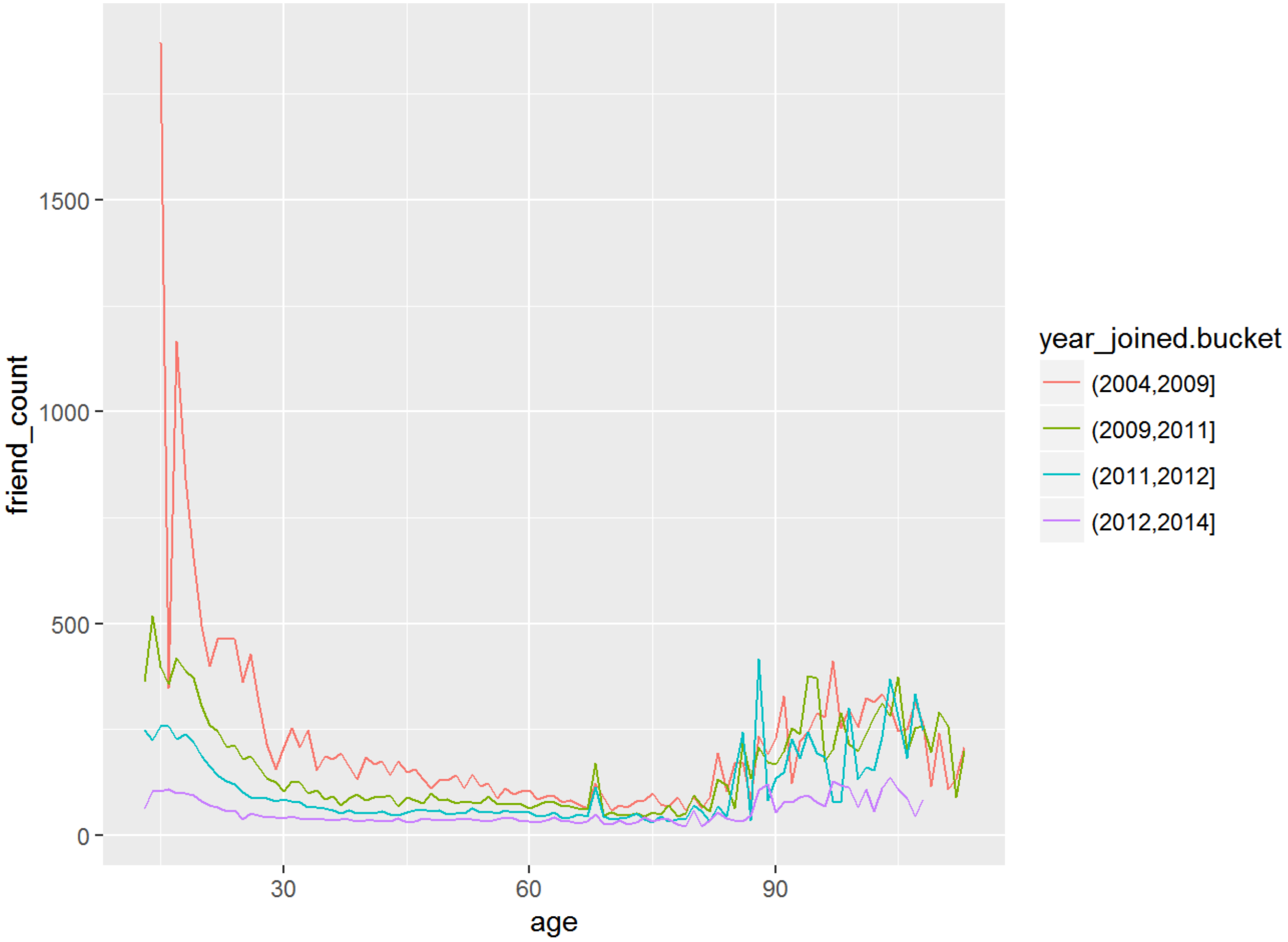
```
table(pf$year_joined.bucket)
```

```
##
## (2004,2009] (2009,2011] (2011,2012] (2012,2014]
##        6669       15308       33366       43658
```

```
table(pf$year_joined.bucket, useNA = "ifany")
```

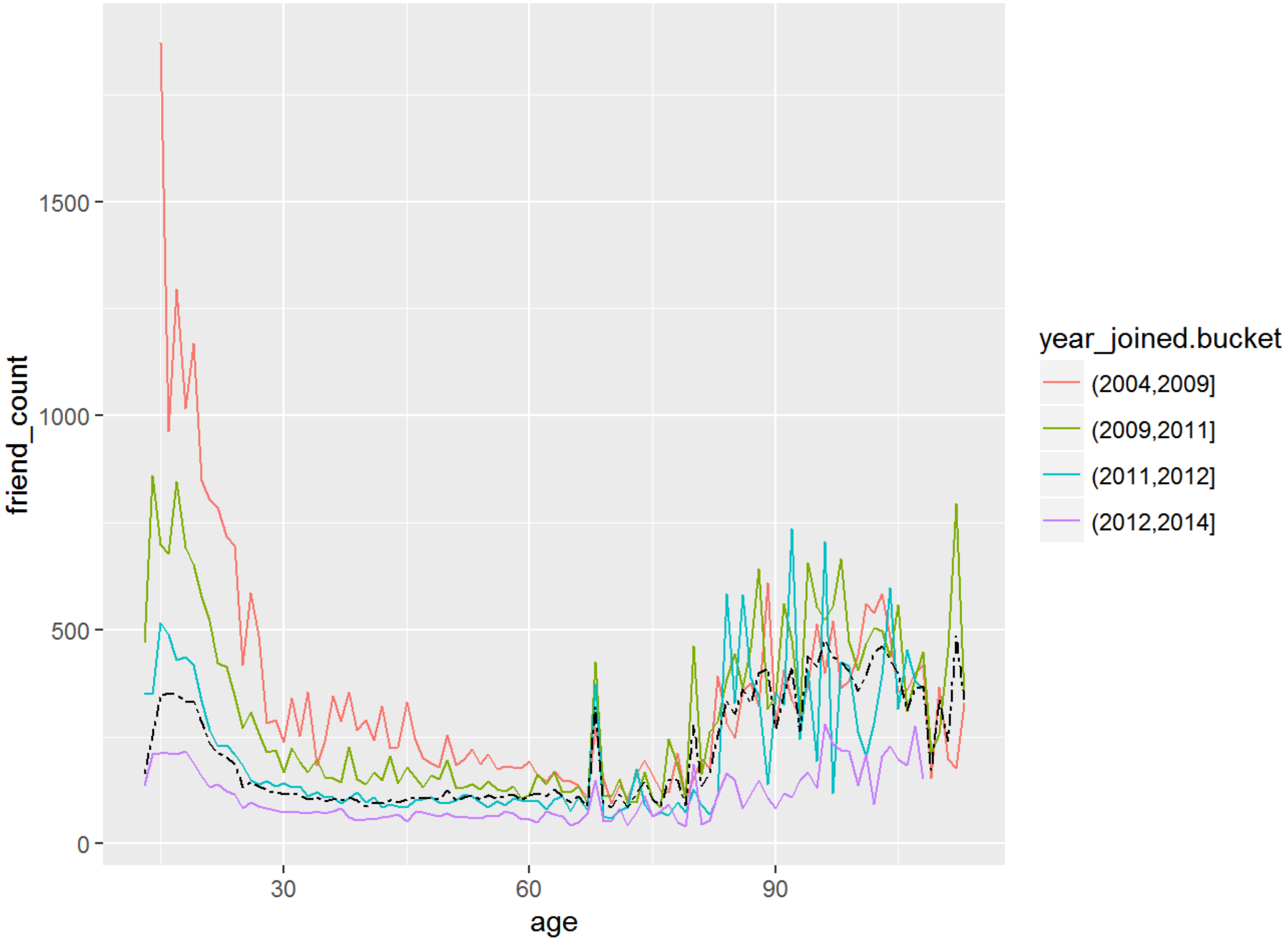
```
##
## (2004,2009] (2009,2011] (2011,2012] (2012,2014]        <NA>
##        6669       15308       33366       43658           2
```

```
ggplot(aes(x = age, y = friend_count),
       data = subset(pf, !is.na(year_joined.bucket))) +
  geom_line(aes(color = year_joined.bucket) ,stat = "summary", fun.y = median)
```



Plot the Grand Mean

```
ggplot(aes(x = age, y = friend_count),
       data = subset(pf, !is.na(year_joined.bucket))) +
  geom_line(aes(color = year_joined.bucket) ,stat = "summary", fun.y = mean) +
  geom_line(linetype = 6 ,stat = "summary", fun.y = mean)
```



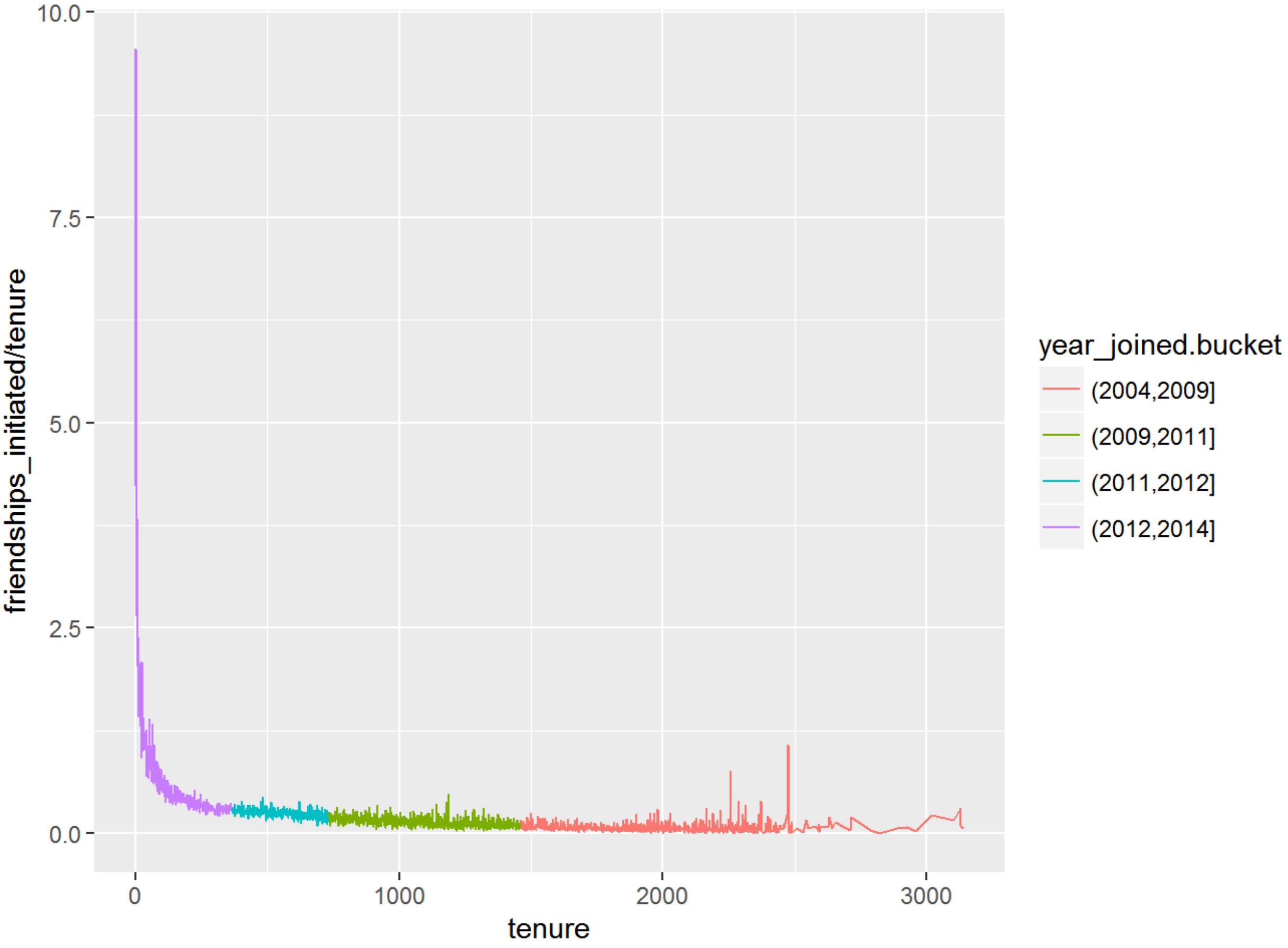
Friending Rate

```
# Friend rate per day
# subset  tenure more than 1 day
with(subset(pf, tenure >= 1), summary(friend_count / tenure))
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max. 
##   0.0000   0.0775   0.2205   0.6096   0.5658 417.0000
```

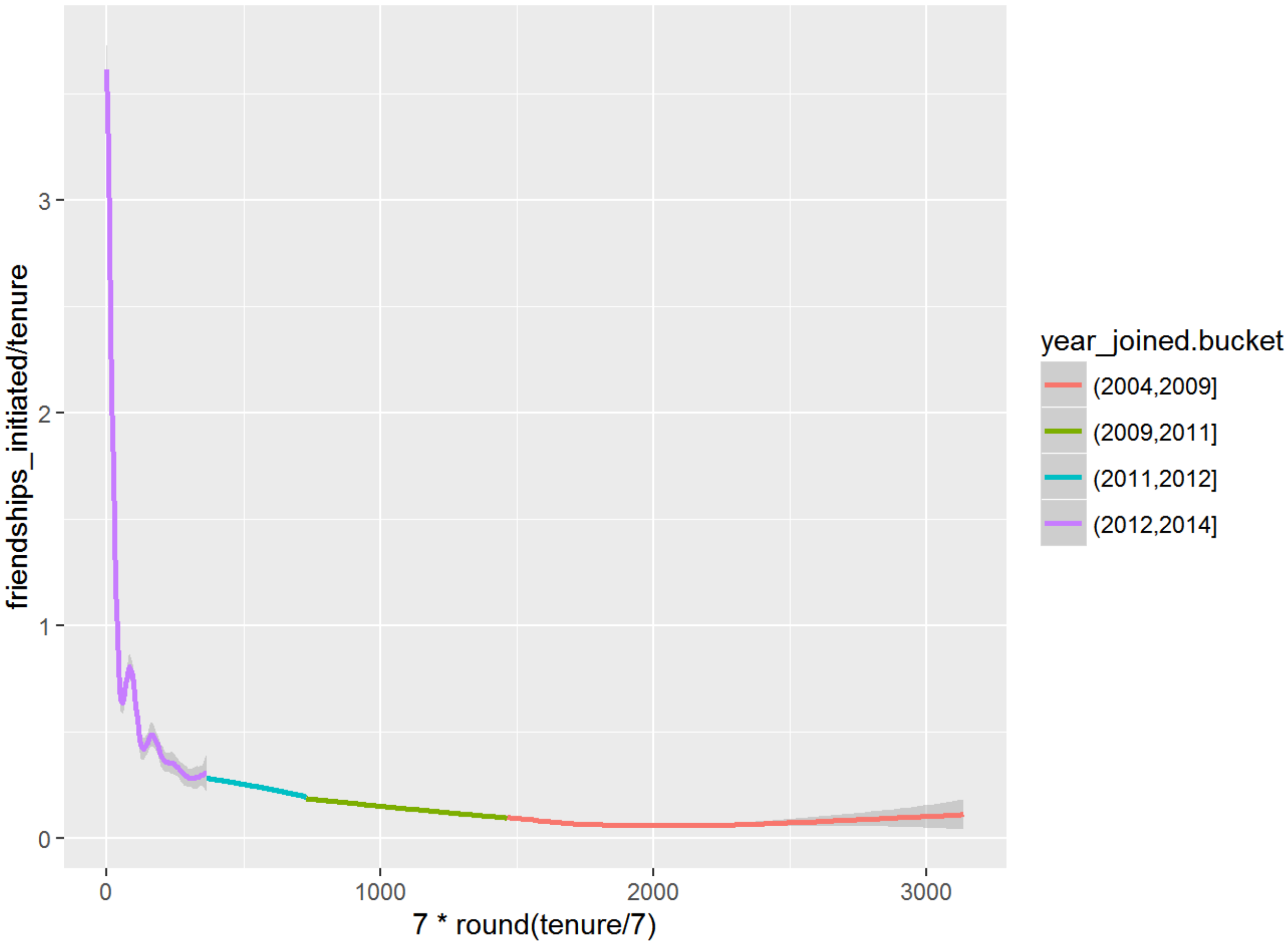
Friendships Initiated

```
ggplot(aes(x = tenure, y = friendships_initiated / tenure),
       data = subset(pf, tenure >= 1)) +
  geom_line(aes(color = year_joined.bucket) ,stat = "summary", fun.y = mean)
```

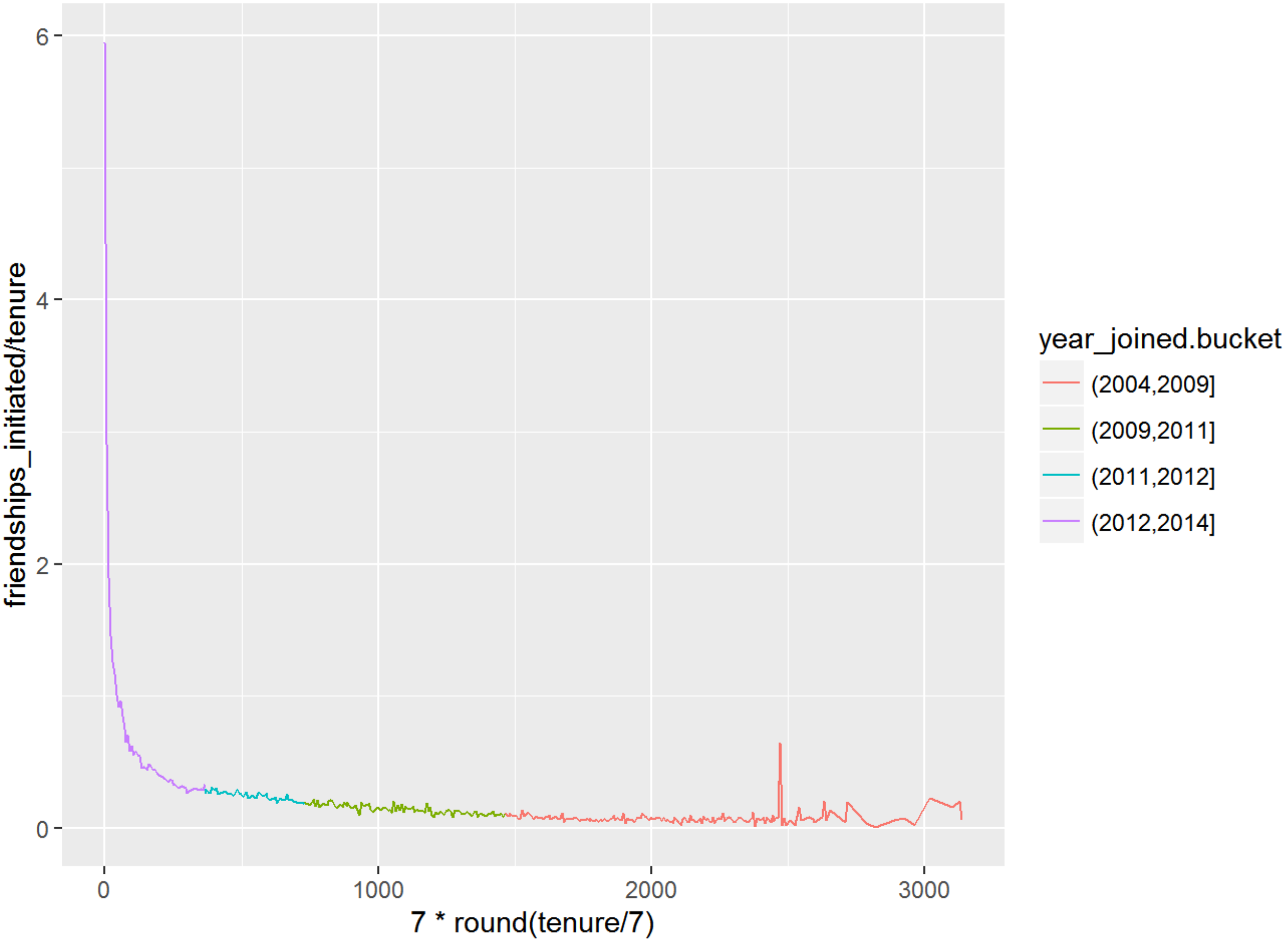


Bias-Variance Tradeoff Revisited

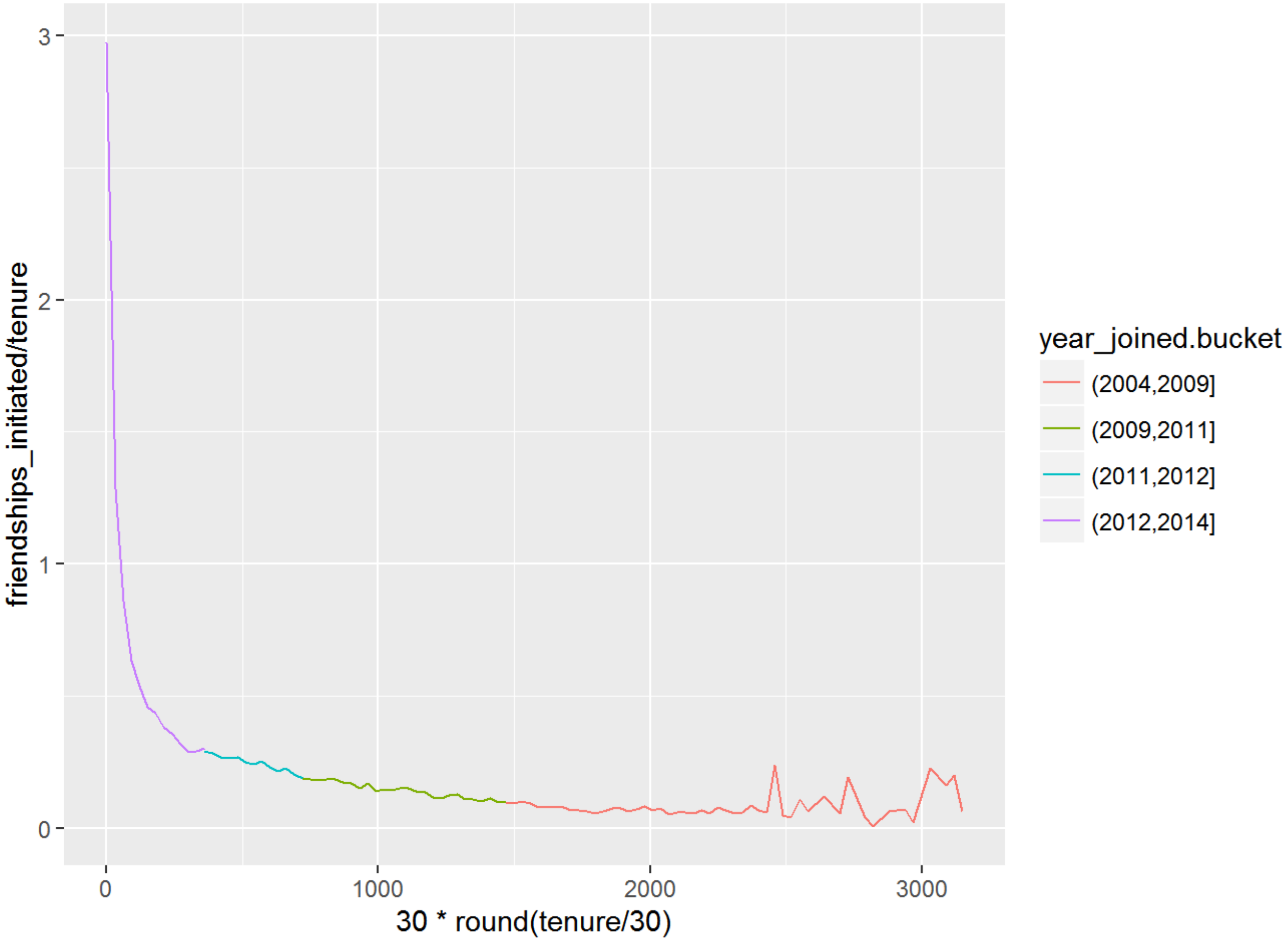
```
ggplot(aes(x = 7 * round(tenure / 7), y = friendships_initiated / tenure),
       data = subset(pf, tenure > 0)) +
  geom_smooth(aes(color = year_joined.bucket))
```



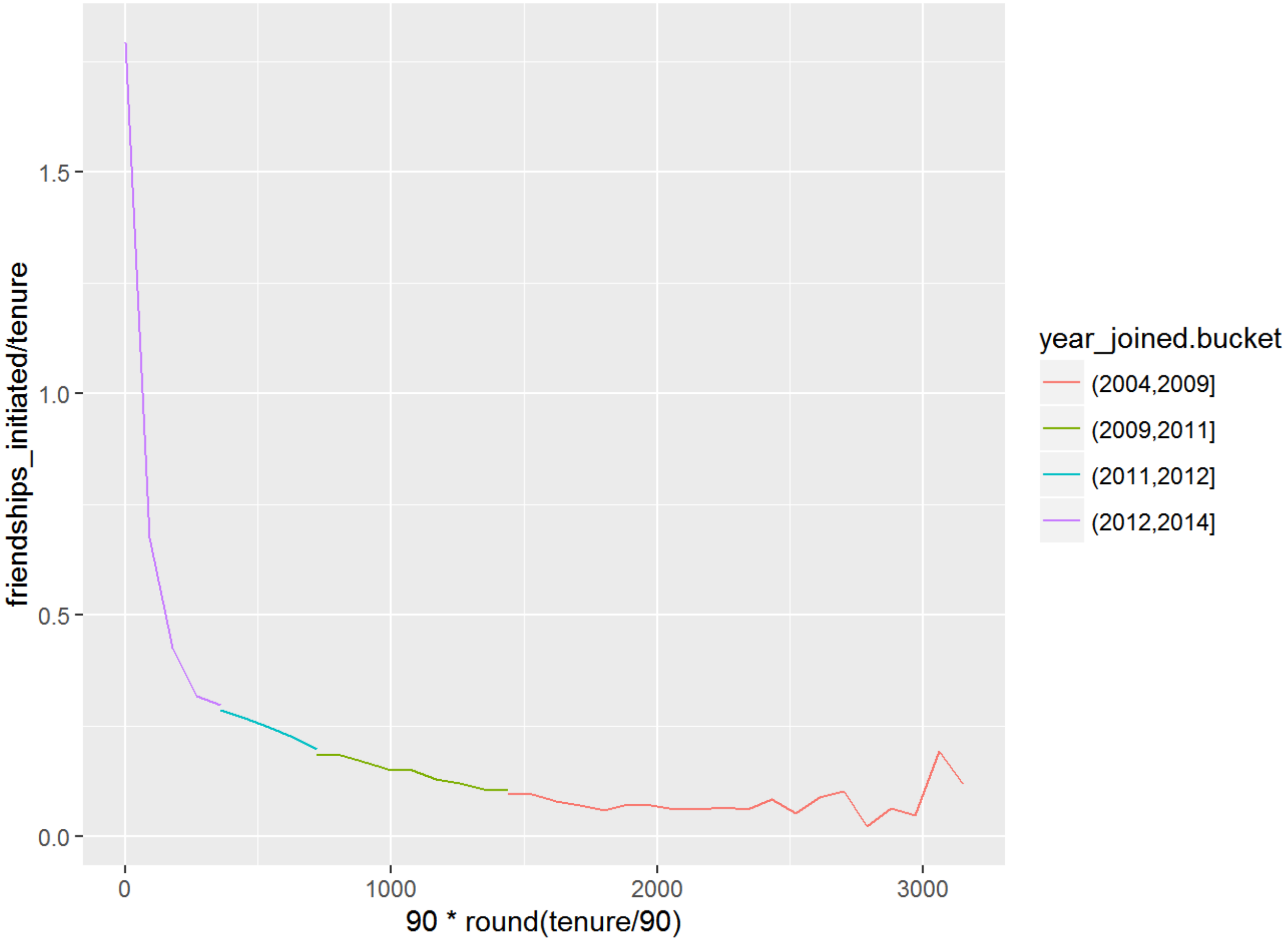
```
ggplot(aes(x = 7 * round(tenure / 7), y = friendships_initiated / tenure),
       data = subset(pf, tenure > 0)) +
  geom_line(aes(color = year_joined.bucket),
            stat = "summary",
            fun.y = mean)
```



```
ggplot(aes(x = 30 * round(tenure / 30), y = friendships_initiated / tenure),
       data = subset(pf, tenure > 0)) +
  geom_line(aes(color = year_joined.bucket),
            stat = "summary",
            fun.y = mean)
```



```
ggplot(aes(x = 90 * round(tenure / 90), y = friendships_initiated / tenure),
       data = subset(pf, tenure > 0)) +
  geom_line(aes(color = year_joined.bucket),
            stat = "summary",
            fun.y = mean)
```



Introducing the Yogurt Data Set

Notes:

Histograms Revisited

```
yo <- read.csv("yogurt.csv")
str(yo)
```

```
## 'data.frame':    2380 obs. of  9 variables:
##  $ obs        : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ id         : int  2100081 2100081 2100081 2100081 2100081 2100081 2100081 2100081
2100081 2100081 ...
##  $ time       : int  9678 9697 9825 9999 10015 10029 10036 10042 10083 10091 ...
##  $ strawberry : int  0 0 0 0 1 1 0 0 0 0 ...
##  $ blueberry  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ pina.colada: int  0 0 0 0 1 2 0 0 0 0 ...
##  $ plain      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ mixed.berry: int  1 1 1 1 1 1 1 1 1 1 ...
##  $ price      : num  59 59 65 65 49 ...
```

```
summary(yo)
```

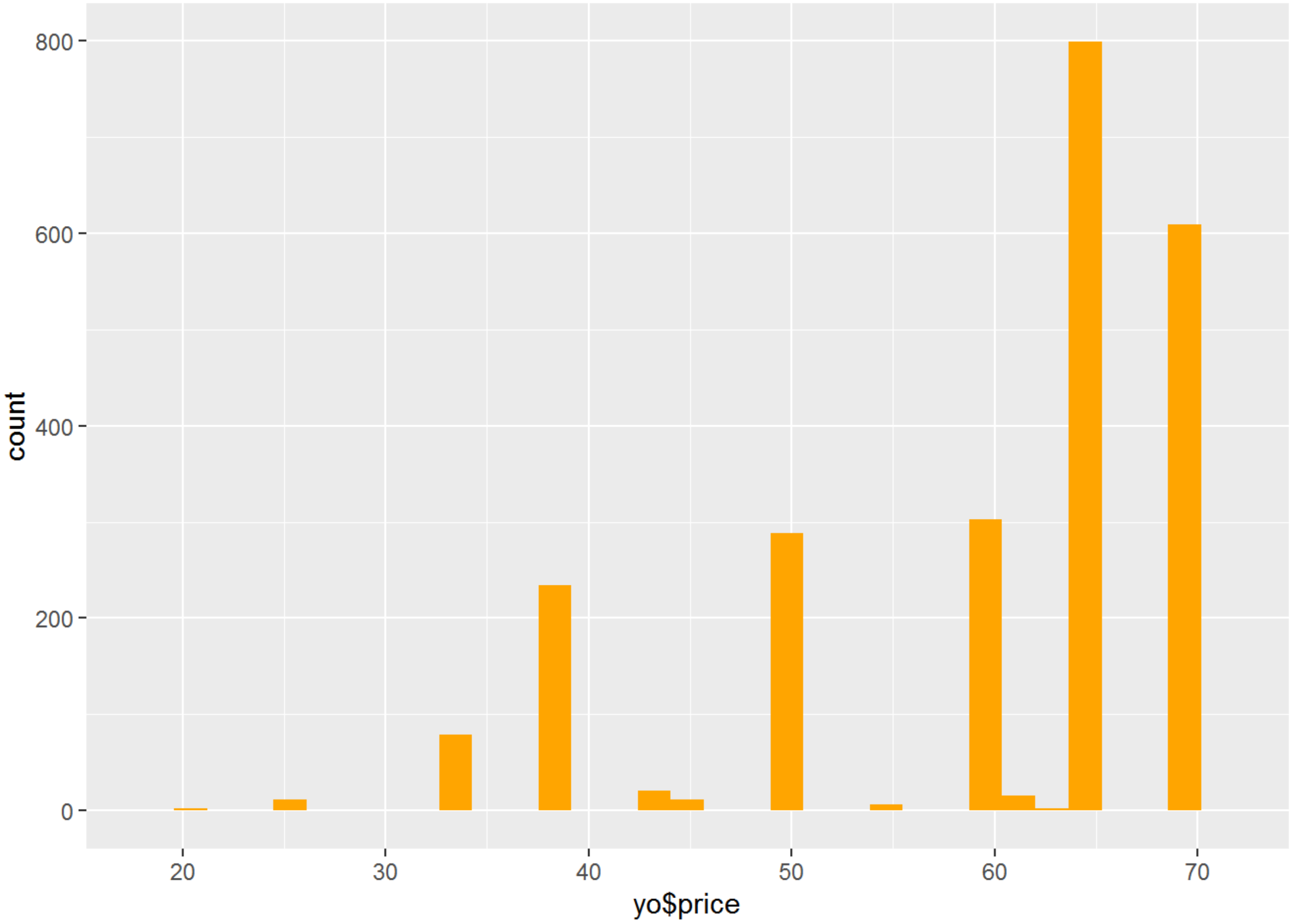
```
##       obs               id               time         strawberry     
##  Min.   :   1.0   Min.   :2100081   Min.   : 9662   Min.   : 0.0000  
##  1st Qu.: 696.5   1st Qu.:2114348   1st Qu.: 9843   1st Qu.: 0.0000  
##  Median :1369.5   Median :2126532   Median :10045   Median : 0.0000  
##  Mean   :1367.8   Mean   :2128592   Mean   :10050   Mean   : 0.6492  
##  3rd Qu.:2044.2   3rd Qu.:2141549   3rd Qu.:10255   3rd Qu.: 1.0000  
##  Max.   :2743.0   Max.   :2170639   Max.   :10459   Max.   :11.0000  
##    blueberry        pina.colada          plain         mixed.berry    
##  Min.   : 0.0000   Min.   : 0.0000   Min.   :0.0000   Min.   :0.0000  
##  1st Qu.: 0.0000   1st Qu.: 0.0000   1st Qu.:0.0000   1st Qu.:0.0000  
##  Median : 0.0000   Median : 0.0000   Median :0.0000   Median :0.0000  
##  Mean   : 0.3571   Mean   : 0.3584   Mean   :0.2176   Mean   :0.3887  
##  3rd Qu.: 0.0000   3rd Qu.: 0.0000   3rd Qu.:0.0000   3rd Qu.:0.0000  
##  Max.   :12.0000   Max.   :10.0000   Max.   :6.0000   Max.   :8.0000  
##      price      
##  Min.   :20.00  
##  1st Qu.:50.00  
##  Median :65.04  
##  Mean   :59.25  
##  3rd Qu.:68.96  
##  Max.   :68.96  
```

```
# convert id to factor
library(ggplot2)
yo$id <- factor(yo$id)
str(yo)
```

```
## 'data.frame':    2380 obs. of  9 variables:
##  $ obs        : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ id         : Factor w/ 332 levels "2100081","2100370",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ time       : int  9678 9697 9825 9999 10015 10029 10036 10042 10083 10091 ...
##  $ strawberry : int  0 0 0 0 1 1 0 0 0 0 ...
##  $ blueberry  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ pina.colada: int  0 0 0 0 1 2 0 0 0 0 ...
##  $ plain      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ mixed.berry: int  1 1 1 1 1 1 1 1 1 1 ...
##  $ price      : num  59 59 65 65 49 ...
```

```
ggplot(data = yo, aes(yo$price)) +
  geom_histogram(fill = "orange")
```

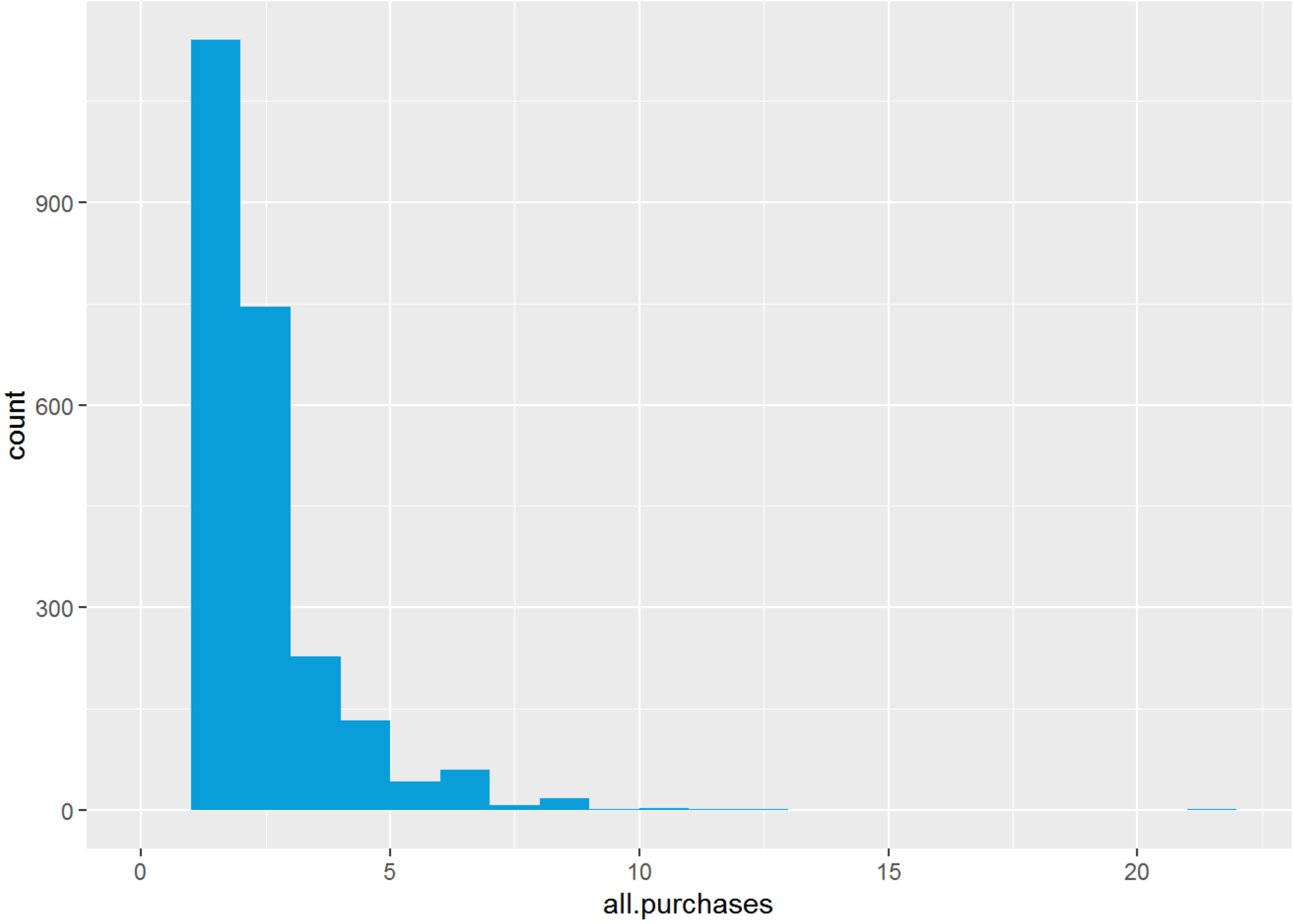
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Number of Purchases

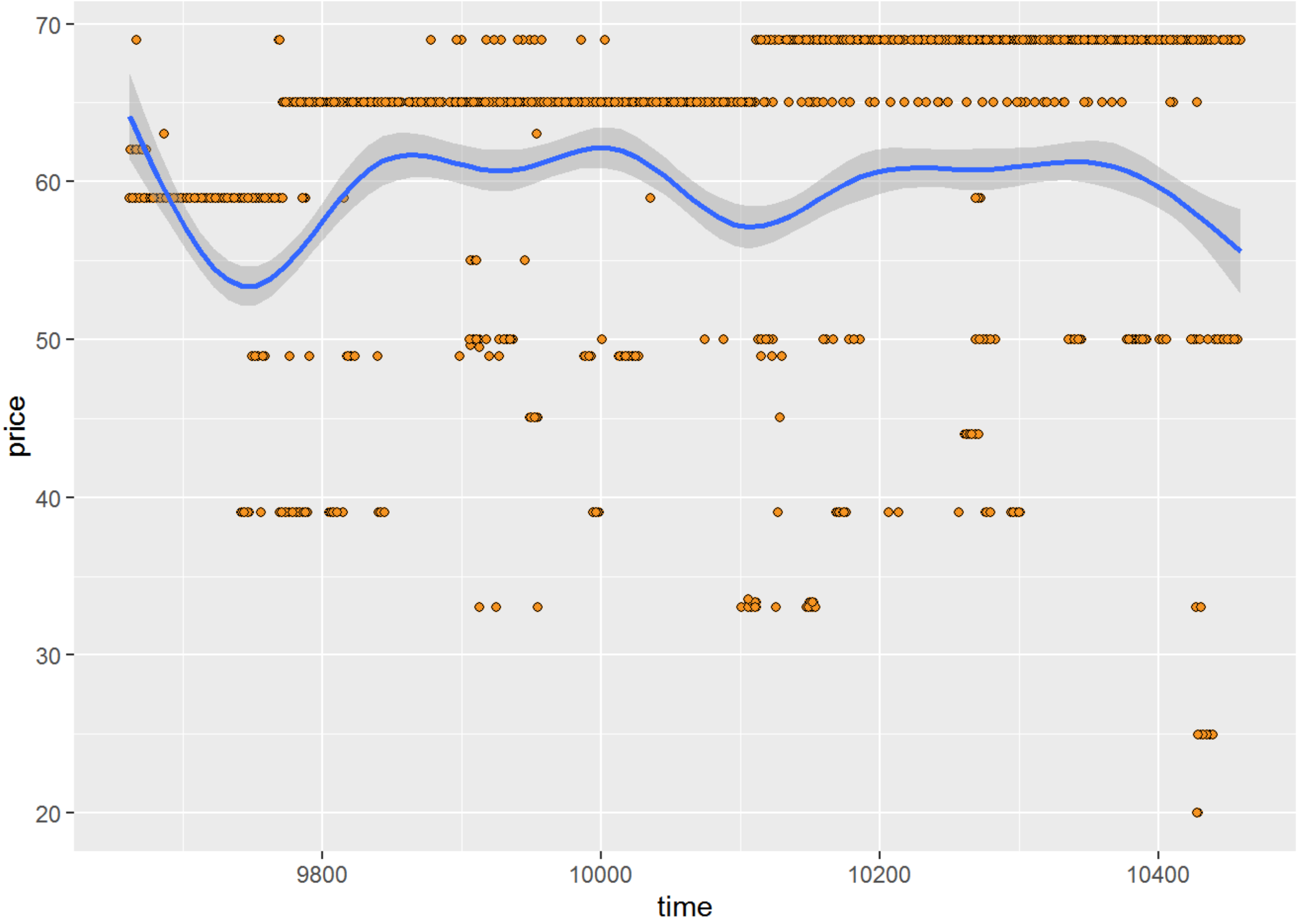
```
yo <- transform(yo, all.purchases = (strawberry + blueberry + pina.colada + plain + mix
ed.berry))

qplot(x = all.purchases, data = yo, binwidth = 1, fill = I("#099DD9"))
```



Prices over Time

```
ggplot(data = yo, aes(x = time, y = price)) +
    geom_point(shape = 21, fill = I("#F79420")) +    # Use hollow circles
    geom_smooth()
```

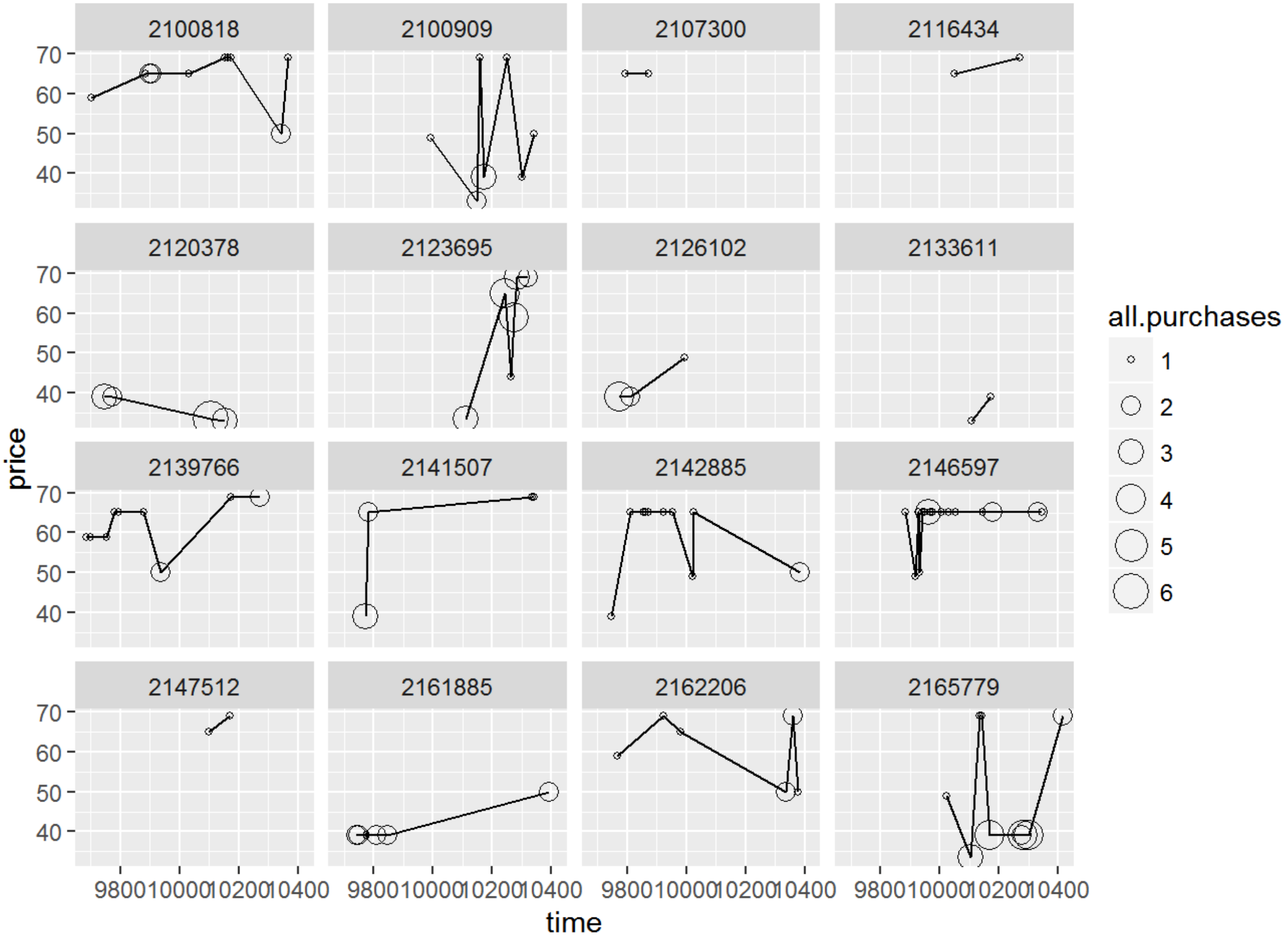


Sampling Observations

Looking at Samples of Households

```
# set the seed number
# new data sample.ids, we sample 16 households
set.seed(6900)
sample.ids <- sample(levels(yo$id), 16)

# %in% for each entry in x, it checks to see whether it is in y.
# This allows us to subset the data so we get all the purchases occasions
# 9for the households in the sample.
# all.purchases makes the circle bigger with more purchases
# pch is the symbol type
ggplot(aes(x = time, y = price),
       data = subset(yo, id %in% sample.ids)) +
  facet_wrap( ~ id) +
  geom_line() +
  geom_point(aes(size = all.purchases), pch = 1)
```



The Limits of Cross Sectional Data

Many Variables

Scatterplot Matrix

Notes:

```
#install.packages("GGally")
library(GGally)
```

```
##
## Attaching package: 'GGally'
##
## The following object is masked from 'package:dplyr':
##
##     nasa
```

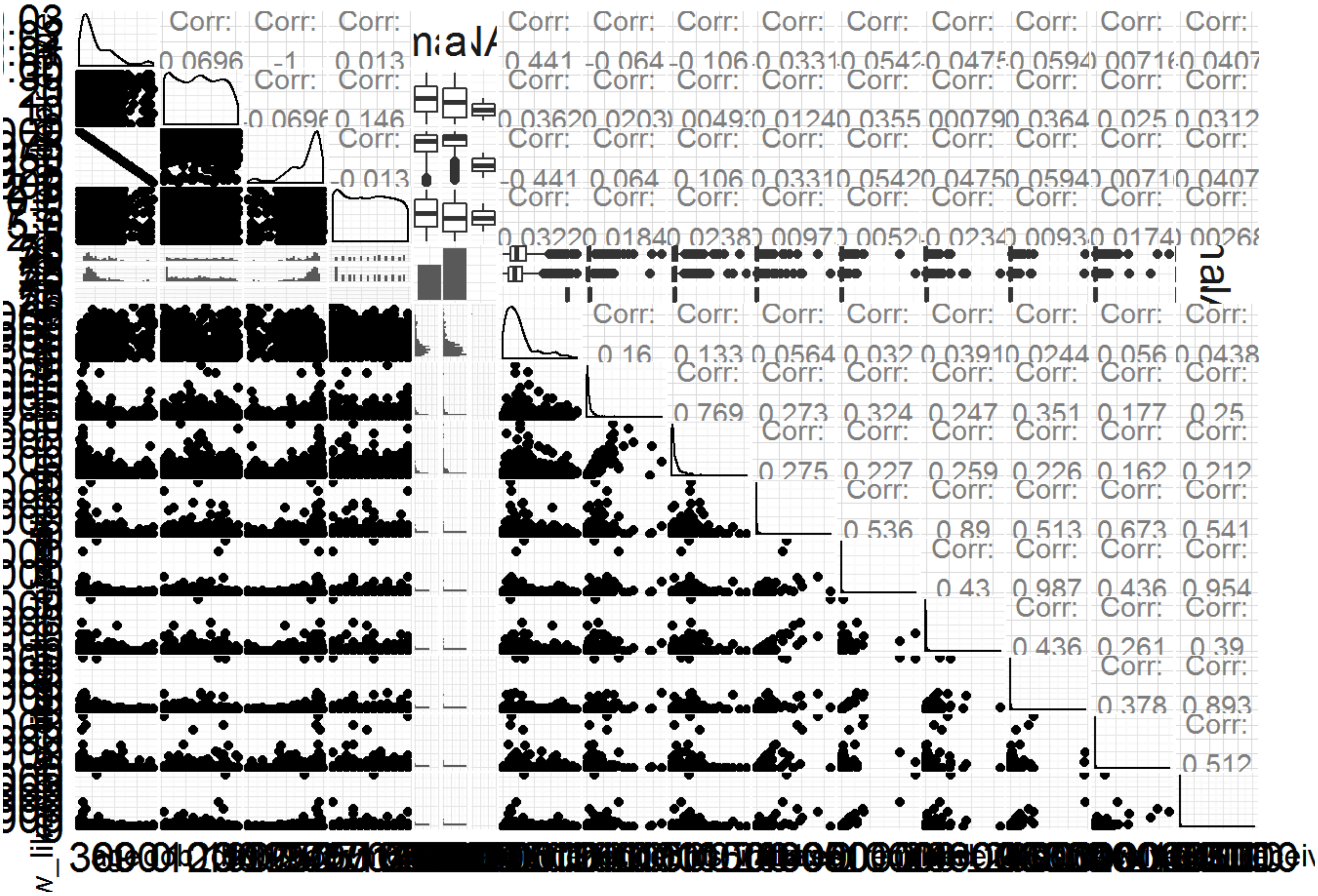
```
theme_set(theme_minimal(20))


set.seed(1836)
pf_subset <- pf[, c(2:15)]
names(pf_subset)
```

```
##  [1] "age"                   "dob_day"
##  [3] "dob_year"              "dob_month"
##  [5] "gender"                "tenure"
##  [7] "friend_count"          "friendships_initiated"
##  [9] "likes"                 "likes_received"
## [11] "mobile_likes"          "mobile_likes_received"
## [13] "www_likes"             "www_likes_received"
```

```
ggpairs(pf_subset[sample.int(nrow(pf_subset), 1000), ])
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
#summary(pf_subset)

#cor(pf_subset, method="kendall")
```

Even More Variables

Heat Maps

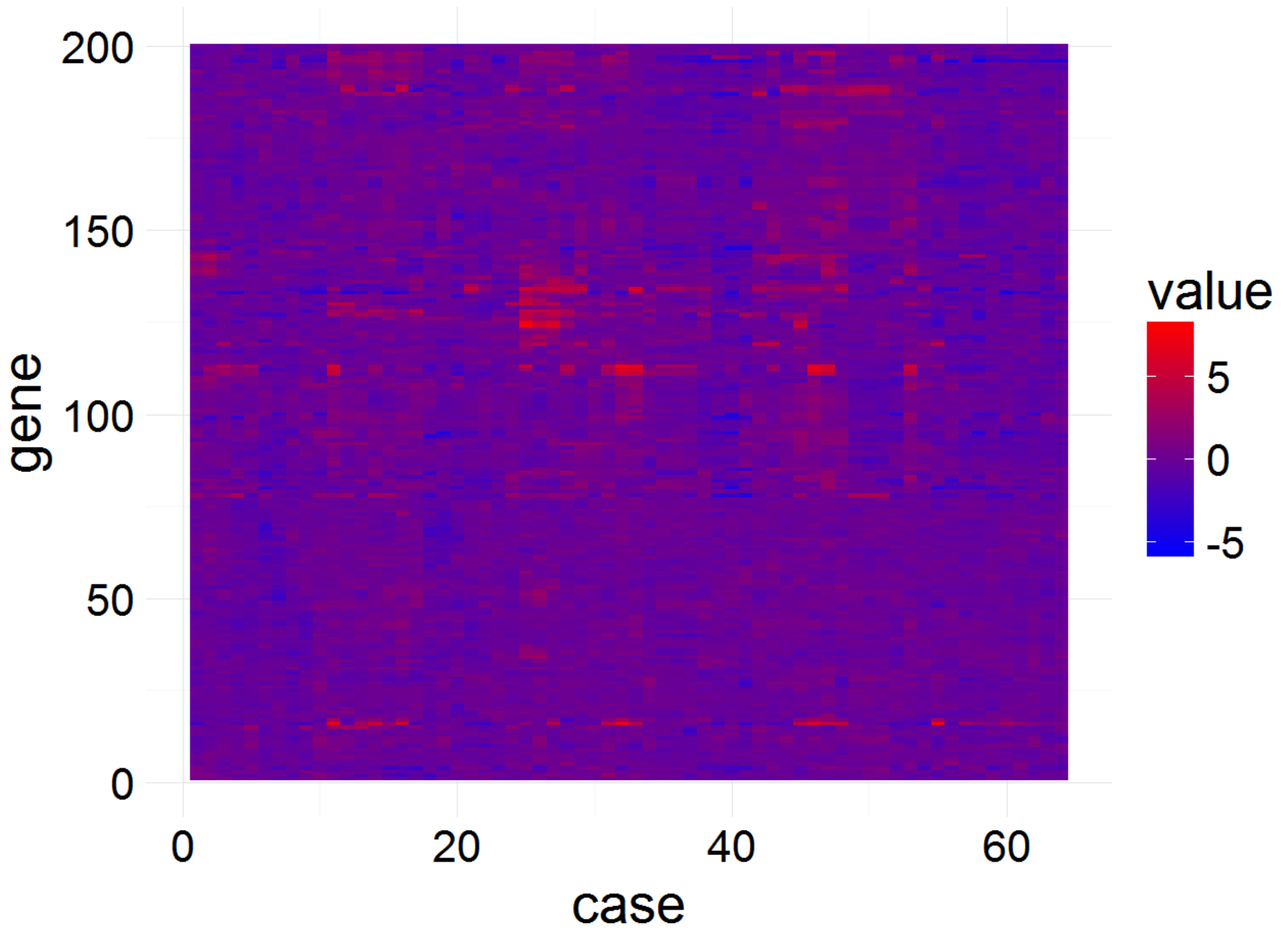
Notes:

```
nci <- read.table("nci.tsv")
# changing the colnames to produce a nicer plot
colnames(nci) <- c(1:64)
```

```
library(reshape2)
nci.long.samp <- melt(as.matrix(nci[1:200,]))
names(nci.long.samp) <- c("gene", "case", "value")
head(nci.long.samp)
```

```
##   gene case  value
## 1    1    1  0.300
## 2    2    1  1.180
## 3    3    1  0.550
## 4    4    1  1.140
## 5    5    1 -0.265
## 6    6    1 -0.070
```

```
library(ggplot2)
ggplot(aes(y = gene, x = case, fill = value),
  data = nci.long.samp) +
  geom_tile() +
  scale_fill_gradientn(colours = colorRampPalette(c("blue", "red"))(100))
```


gene
case
value