# Problem set 8: Worked solutions

Statistics and statistical programming
Northwestern University
MTS 525

Aaron Shaw

November 23, 2020

## Contents



## Import libraries

I'll start by loading some useful libraries...

```
library(openintro)
library(tidyverse)
library(ggfortify)
library(haven)
```

## Part I: Mario kart replication/extension

### Import data and setup

```
data(mariokart)

mariokart
```

```
## # A tibble: 143 x 12
##         id duration n_bids cond  start_pr ship_pr total_pr ship_sp seller_rate
##      <dbl>    <int>  <int> <fct>    <dbl>   <dbl>    <dbl> <fct>         <int>
```

```
##  1 1.50e11        3     20 new       0.99    4        51.6 standa~        1580
##  2 2.60e11        7     13 used      0.99    3.99     37.0 firstC~         365
##  3 3.20e11        3     16 new       0.99    3.5      45.5 firstC~         998
##  4 2.80e11        3     18 new       0.99    0        44   standa~           7
##  5 1.70e11        1     20 new       0.01    0        71   media            820
##  6 3.60e11        3     19 new       0.99    4        45   standa~      270144
##  7 1.20e11        1     13 used      0.01    0        37.0 standa~        7284
##  8 3.00e11        1     15 new       1       2.99     54.0 upsGro~        4858
##  9 2.00e11        3     29 used      0.99    4        47   priori~          27
## 10 3.30e11        7      8 used     20.0     4        50   firstC~         201
## # ... with 133 more rows, and 3 more variables: stock_photo <fct>,
## #   wheels <int>, title <fct>
```

To make things a bit easier to manage, I'll select the variables I want to use in the analysis and do some cleanup. Note that I conver the `cond_new` and `stock_photo` variables to logical first (using boolean comparisons) and also coerce them to be numeric values (using `as.numeric()`). This results in 1/0 values corresponding to the observations shown/described in Figure 9.13 and 9.14 on p. 365 of the textbook.

```
mariokart <- mariokart %>%
  select(
    price = total_pr, cond_new = cond, stock_photo, duration, wheels
  ) %>%
  mutate(
    cond_new = as.numeric(cond_new == "new"),
    stock_photo = as.numeric(stock_photo == "yes")
  )

mariokart
```

```
## # A tibble: 143 x 5
##    price cond_new stock_photo duration wheels
##    <dbl>    <dbl>       <dbl>    <int>  <int>
##  1  51.6        1           1        3      1
##  2  37.0        0           1        7      1
##  3  45.5        1           0        3      1
##  4  44          1           1        3      1
##  5  71          1           1        1      2
##  6  45          1           1        3      0
##  7  37.0        0           1        1      0
##  8  54.0        1           1        1      2
##  9  47          0           1        3      1
## 10  50          0           0        7      1
## # ... with 133 more rows
```

Now let's look at the variables in our model. Summary statistics for each, univariate density plots for the continuous measures, and some bivariate plots w each predictor and the outcome variable.

```
summary(mariokart)
```

```
##      price           cond_new       stock_photo        duration
##  Min.   : 28.98   Min.   :0.0000   Min.   :0.0000   Min.   : 1.000
##  1st Qu.: 41.17   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.: 1.000
##  Median : 46.50   Median :0.0000   Median :1.0000   Median : 3.000
##  Mean   : 49.88   Mean   :0.4126   Mean   :0.7343   Mean   : 3.769
##  3rd Qu.: 53.99   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.: 7.000
##  Max.   :326.51   Max.   :1.0000   Max.   :1.0000   Max.   :10.000
##      wheels
```

```
##  Min.   :0.000
##  1st Qu.:0.000
##  Median :1.000
##  Mean   :1.147
##  3rd Qu.:2.000
##  Max.   :4.000
```

```
sd(mariokart$price)
```

```
## [1] 25.68856
```

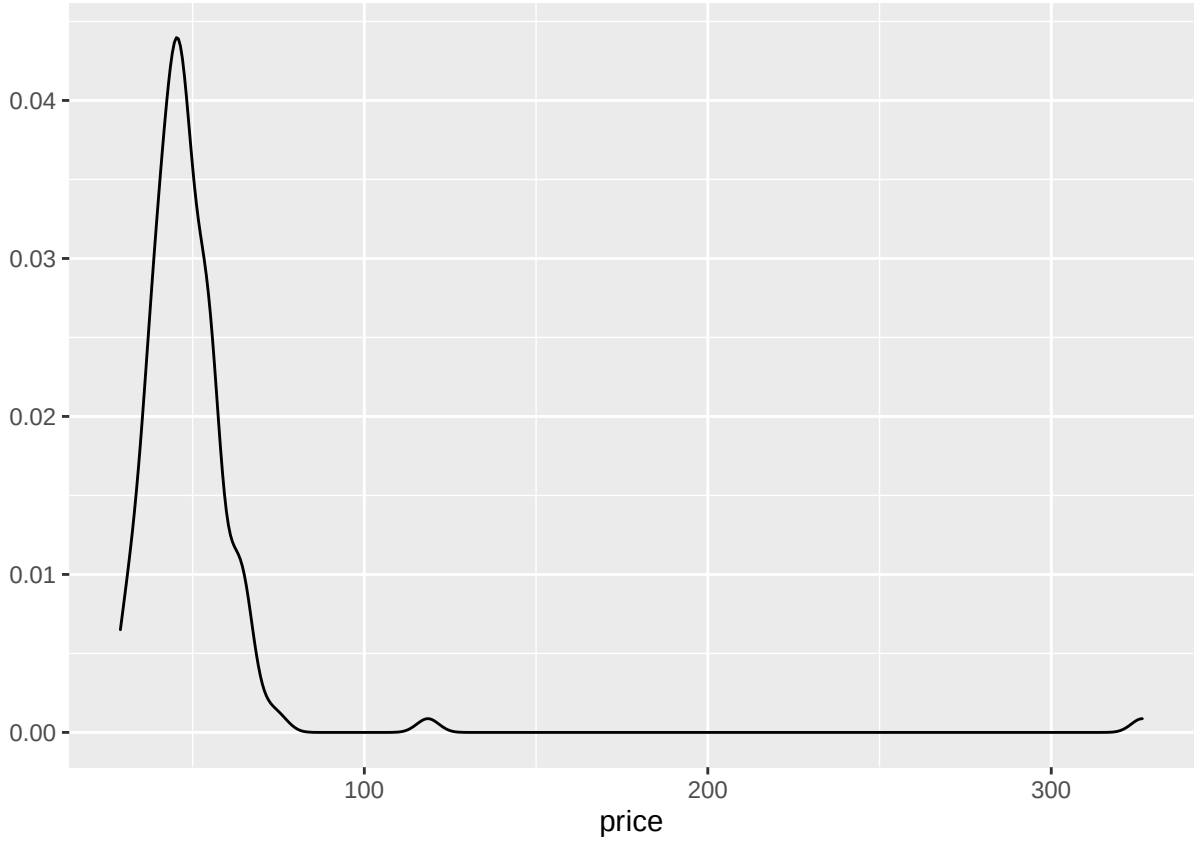
```
sd(mariokart$duration)
```

```
## [1] 2.585693
```

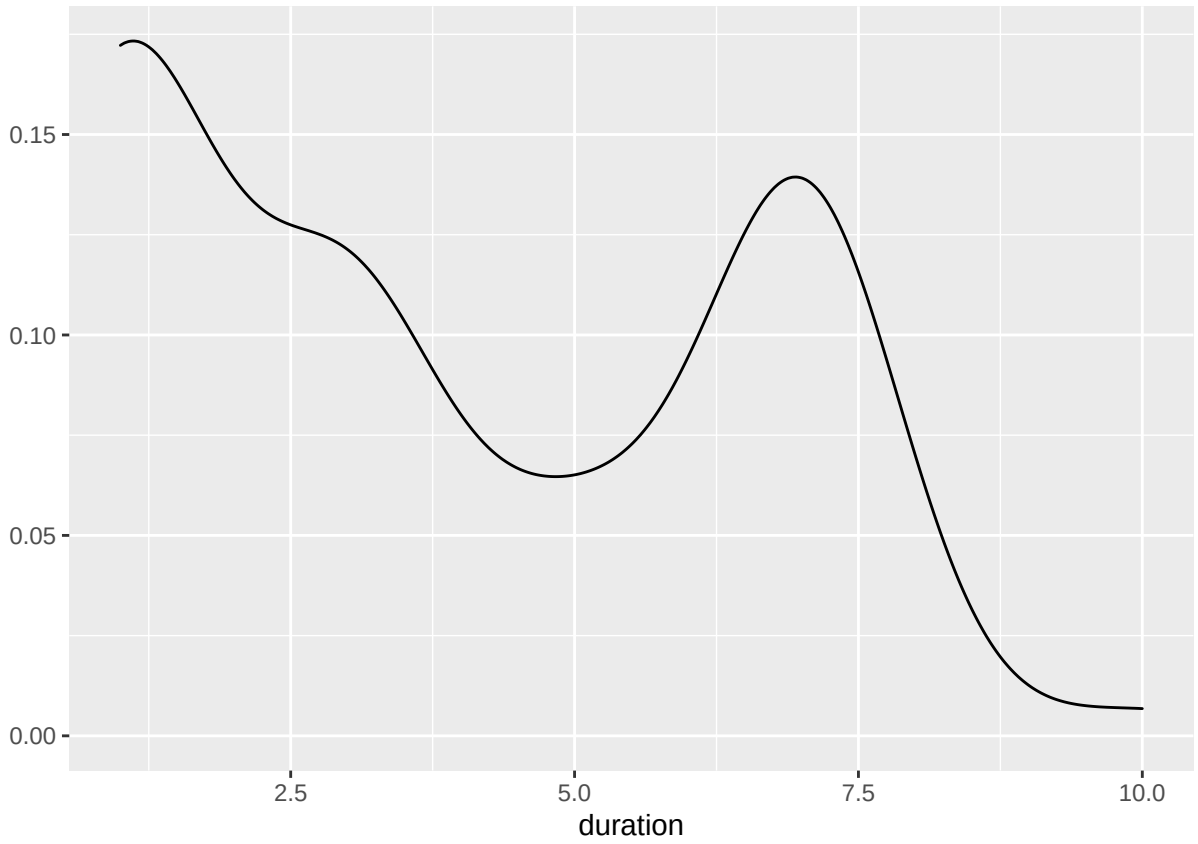
```
sd(mariokart$wheels)
```

```
## [1] 0.8471829
```

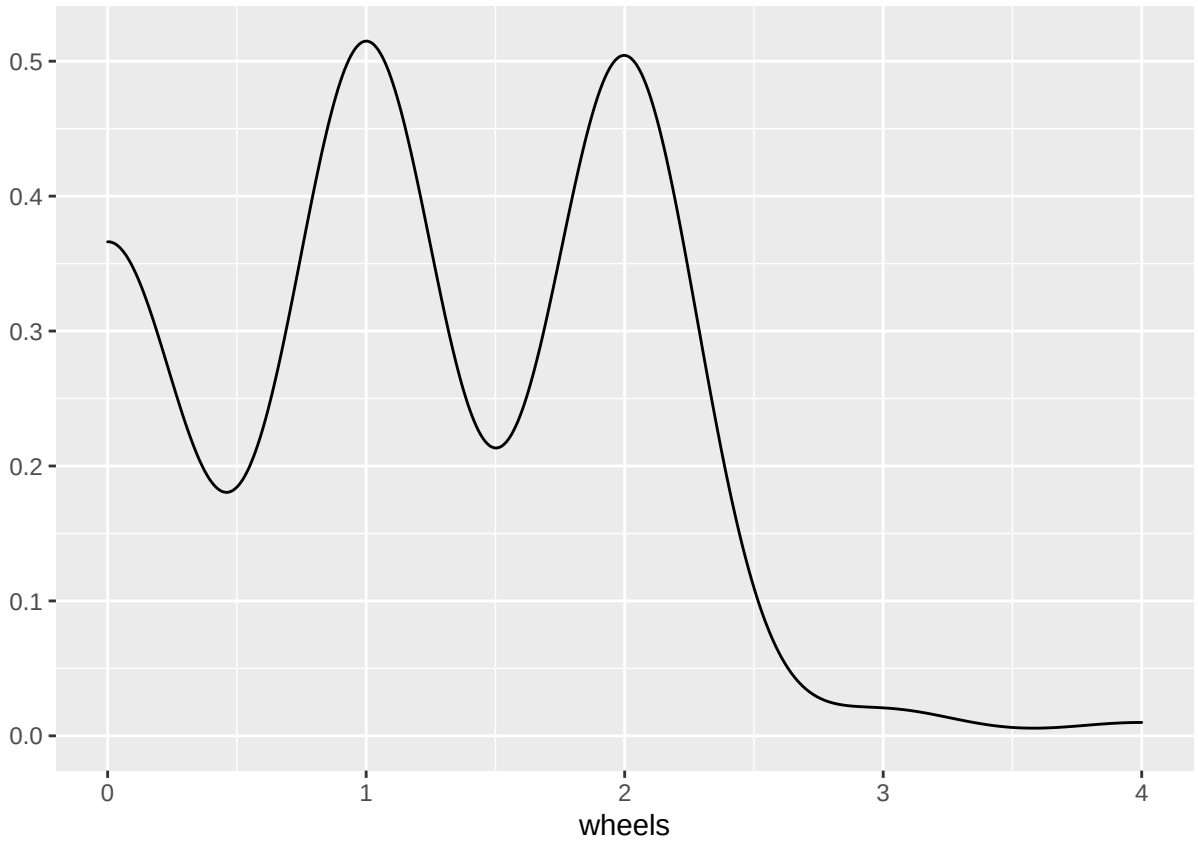
```
qplot(data = mariokart, x = price, geom = "density")
```




```
qplot(data = mariokart, x = duration, geom = "density")
```

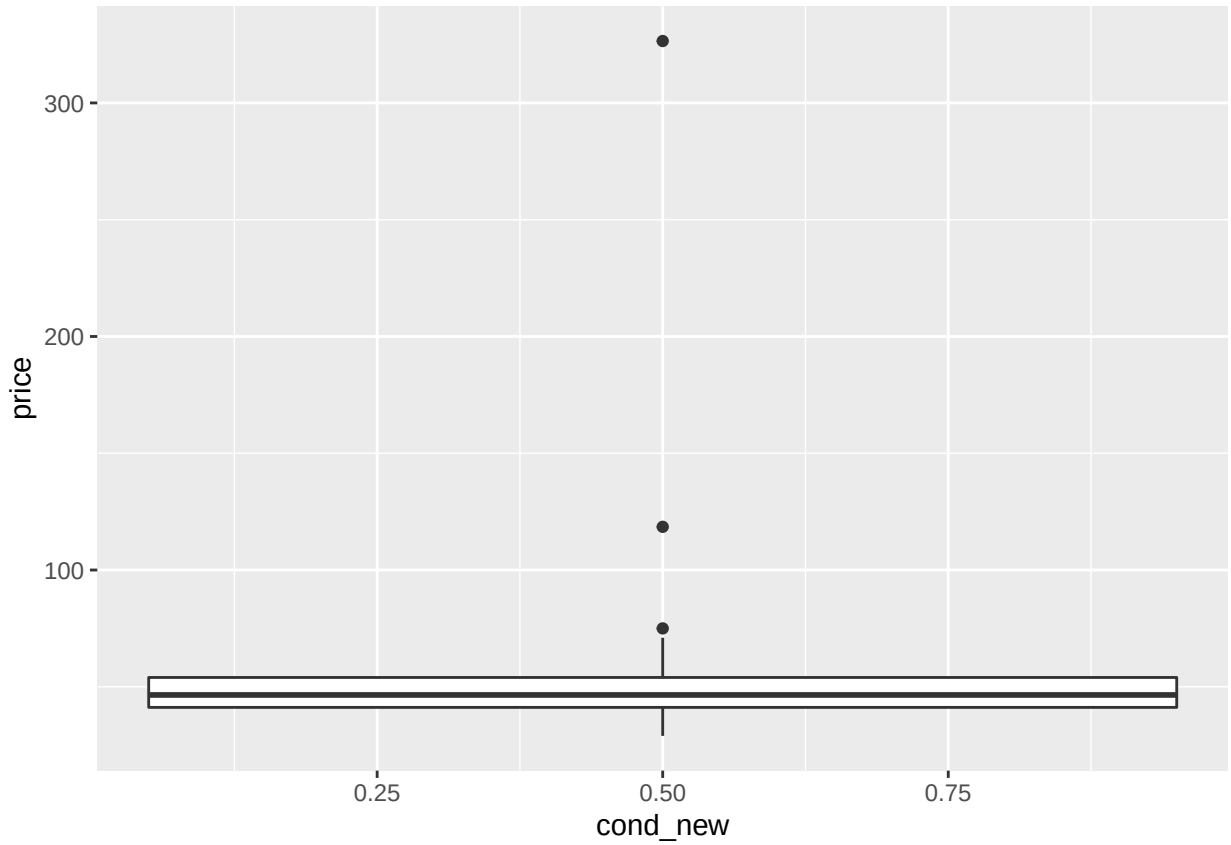



```
qplot(data = mariokart, x = wheels, geom = "density")
```



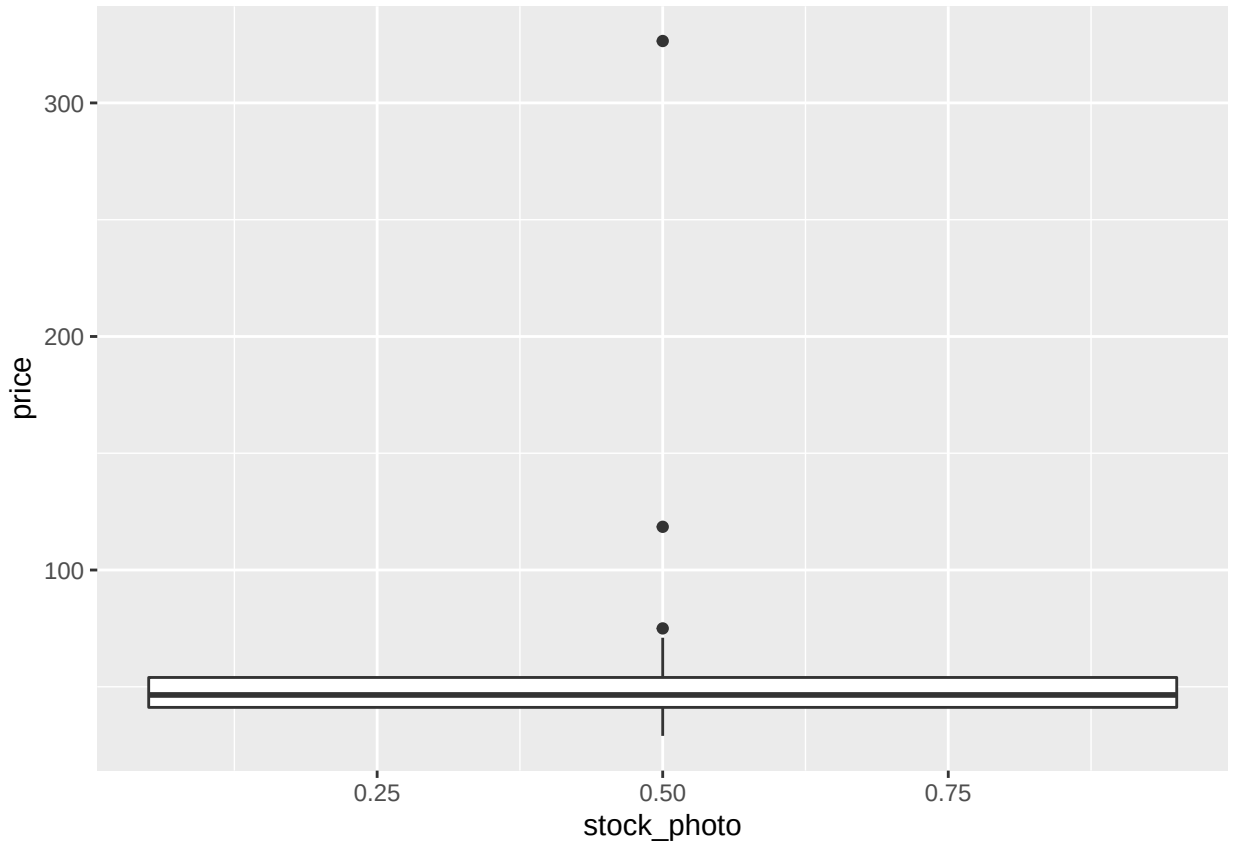

```
ggplot(data = mariokart, aes(cond_new, price)) +
  geom_boxplot()
```

```
## Warning: Continuous x aesthetic -- did you forget aes(group=...)?
```

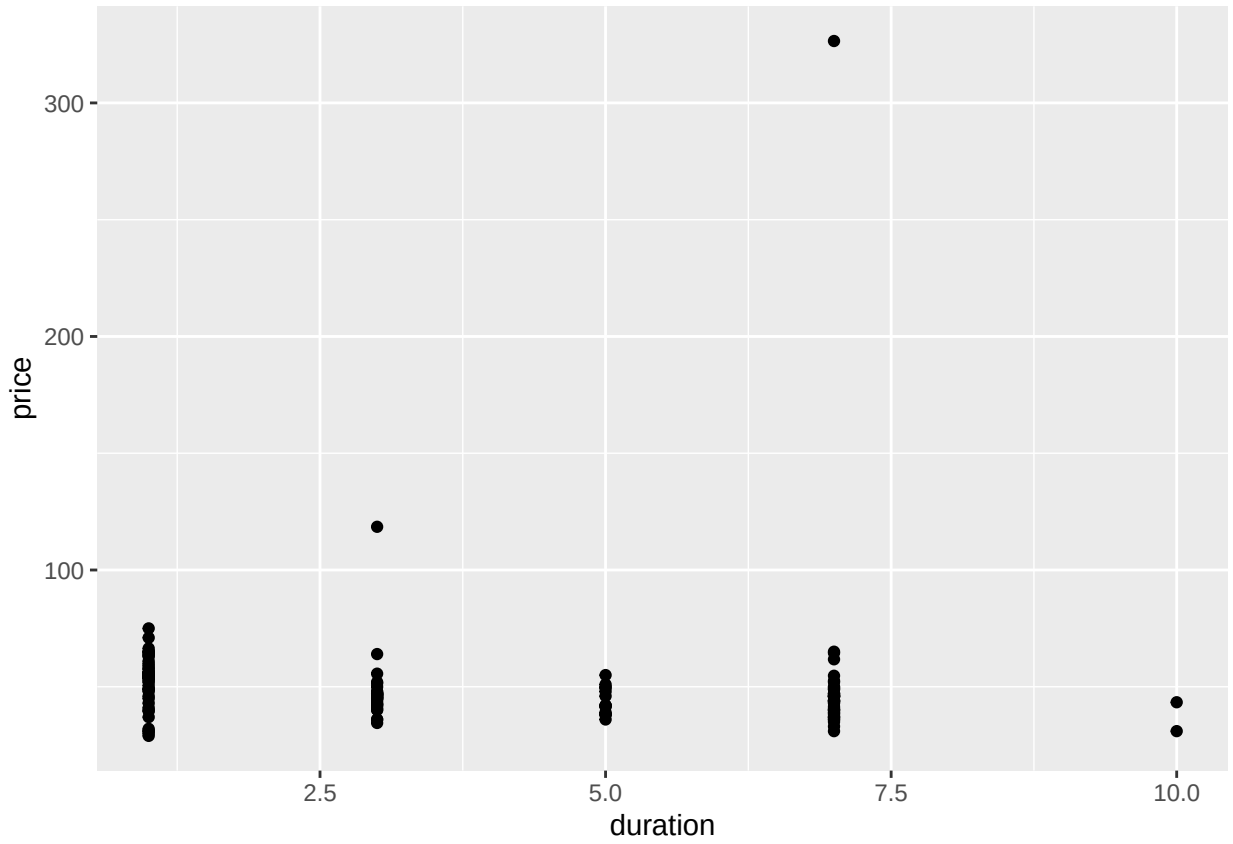



```
ggplot(data = mariokart, aes(stock_photo, price)) +
  geom_boxplot()
```

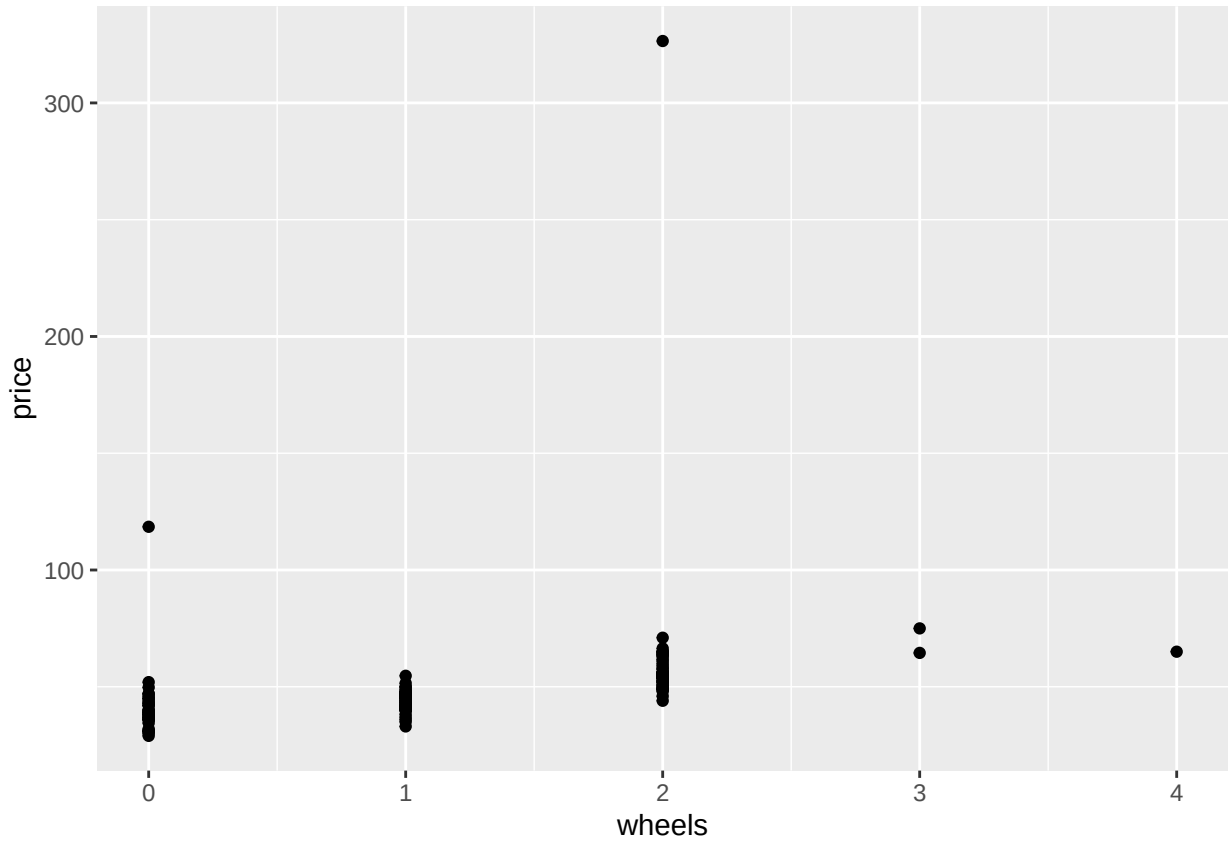
```
## Warning: Continuous x aesthetic -- did you forget aes(group=...)?
```




```
ggplot(data = mariokart, aes(duration, price)) +
  geom_point()
```




```
ggplot(data = mariokart, aes(wheels, price)) +
  geom_point()
```




I'm also going to calculate correlation coefficients for all of the variables. We can discuss what to make of this in class:

```
cor(mariokart)
```

```
##                   price   cond_new stock_photo    duration      wheels
## price        1.00000000  0.1273624 -0.08987876 -0.04123545  0.32998375
## cond_new     0.12736236  1.0000000  0.37554707 -0.48174393  0.42629801
## stock_photo -0.08987876  0.3755471  1.00000000 -0.36722999  0.06714198
## duration    -0.04123545 -0.4817439 -0.36722999  1.00000000 -0.29947345
## wheels       0.32998375  0.4262980  0.06714198 -0.29947345  1.00000000
```

## Replicate model results

The description of the model

```
model <- lm(price ~ cond_new + stock_photo + duration + wheels, data = mariokart)
summary(model)
```

```
##
## Call:
## lm(formula = price ~ cond_new + stock_photo + duration + wheels,
##     data = mariokart)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -19.485  -6.511  -2.530   1.836 263.025
##
## Coefficients:
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  40.9385     7.3607   5.562 1.34e-07 ***
## cond_new      2.5816     5.2272   0.494 0.622183
## stock_photo  -6.7542     5.1729  -1.306 0.193836
## duration      0.3788     0.9388   0.403 0.687206
## wheels        9.9476     2.7184   3.659 0.000359 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 24.4 on 138 degrees of freedom
## Multiple R-squared:  0.1235, Adjusted R-squared:  0.09808
## F-statistic:  4.86 on 4 and 138 DF,  p-value: 0.001069
```

While I'm looking at that, I'll go ahead and calculate a confidence interval around the only parameter for which the model rejects the null hypothesis (`wheels`):

```
confint(model, "wheels")
```

```
##           2.5 %   97.5 %
## wheels 4.572473 15.32278
```

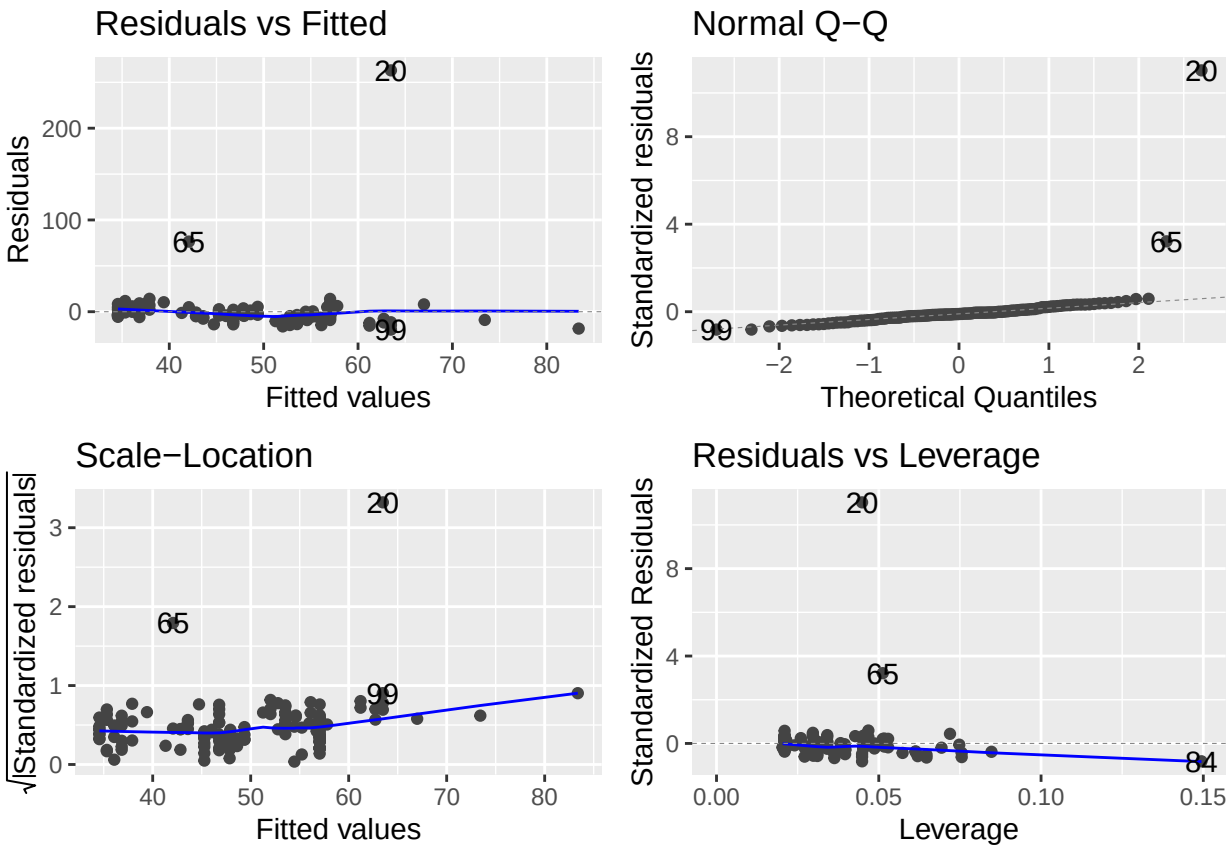
Overall, this resembles the model results in Figure 9.15, but notice that it's different in a number of ways! Without more information from the authors of the textbook it's very hard to determine exactly where or why the differences emerge.

## Assess model fit/assumptions

I've already generated a bunch of univariate and bivariate summaries and plots. Let's inspect the residuals more closely to see what else we can learn. I'll use the `autoplot()` function from the `ggfortify` package to help me do this.

```
autoplot(model)
```

```
## Warning: `arrange_()` is deprecated as of dplyr 0.7.0.
## Please use `arrange()` instead.
## See vignette('programming') for more help
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```




Overall, there are a number of issues with this model fit that I'd want to mention/consider: * The distribution of the dependent variable (`price`) is very skewed, with two extreme outliers. I'd recommend trying some transformations to see if it would look more appropriate for a linear regression and/or inspecting the cases that produced the outlying values more closely to understand what's happening there. * The plots of the residuals reveal those same two outlying points are also outliers with respect to the line of best fit. That said, they are not exerting huge amounts of leverage on the estimates, so it's possible that the estimates from the fitted model wouldn't change much without those two points. Indeed, based on the degrees of freedom reported in Figure 9.15 (136) vs. the number reported in our version of the model (138) my best guess is that the textbook authors silently dropped those two outlying observations from their model!

More out of curiosity than anything, I'll create a version of the model that drops the two largest values of price. From the plots, I can see that those two are the only ones above $100, so I'll use that information here:

```
summary(lm(price ~ cond_new + stock_photo + duration + wheels, data = mariokart[mariokart$price < 100, ]
```

```
##
## Call:
## lm(formula = price ~ cond_new + stock_photo + duration + wheels,
##     data = mariokart[mariokart$price < 100, ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.3788  -2.9854  -0.9654   2.6915  14.0346
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 36.21097    1.51401  23.917  < 2e-16 ***
## cond_new     5.13056    1.05112   4.881 2.91e-06 ***
```

```
## stock_photo  1.08031    1.05682   1.022    0.308
## duration    -0.02681    0.19041  -0.141    0.888
## wheels       7.28518    0.55469  13.134  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.901 on 136 degrees of freedom
## Multiple R-squared:  0.719,  Adjusted R-squared:  0.7108
## F-statistic: 87.01 on 4 and 136 DF,  p-value: < 2.2e-16
```

What do you know. That was it.

### Interpret some results

The issues above notwithstanding, we can march ahead and interpret the model results. Here are some general comments and some specifically focused on the `cond_new` and `stock_photo` variables: * Overall, the linear model regressing total auction price on condition, stock photo, duration, and number of Wii wheels shows evidence of a positive, significant relationship between number of wheels and price. According to this model fit, an increase of 1 wheel is associated with a total auction price increase of $10 with the 95% confindence interval of ($4.57-$15.32). * The point estimate for selling a new condition game is positive, but with a large standard error. As a result, the model fails to reject the null of no association and provides no evidence of any relationship between the game condition and auction price. * The point estimate for including a stock photo is negative, but again, the standard error is very large and the model fails to reject the null hypothesis. There is no evidence of any relationship between including a stock photo and the final auction price.

### Recommendations

Based on this model result, I'd recommend the prospective vendor of a **used** copy of the game not worry about it too much unless they can get their hands on some extra Wii wheels, since it seems like the number of Wii wheels explains variations in the auction price outcome more than anything else they can control (such as whether or not a stock photo is included).

## Part II: Hypothetical study

### Import and explore

I'll start off by just importing things and summarizing the different variables we care about here:

```
grads <- readRDS(url("https://communitydata.science/~ads/teaching/2020/stats/data/week_11/grads.rds"))

summary(grads)
```

```
##        id              cohort         gpa               income
##  Min.   :   1.0   cohort_01:142   Min.   : 0.01512   Min.   :15.56
##  1st Qu.: 462.2   cohort_02:142   1st Qu.:22.56107   1st Qu.:41.07
##  Median : 923.5   cohort_03:142   Median :47.59445   Median :52.59
##  Mean   : 923.5   cohort_04:142   Mean   :47.83510   Mean   :54.27
##  3rd Qu.:1384.8   cohort_05:142   3rd Qu.:71.81078   3rd Qu.:67.28
##  Max.   :1846.0   cohort_06:142   Max.   :99.69468   Max.   :98.29
##                   (Other)  :994
```

```
table(grads$cohort)
```

```
##
## cohort_01 cohort_02 cohort_03 cohort_04 cohort_05 cohort_06 cohort_07 cohort_08
##       142       142       142       142       142       142       142       142
```

```
## cohort_09 cohort_10 cohort_11 cohort_12 cohort_13
##       142       142       142       142       142
```

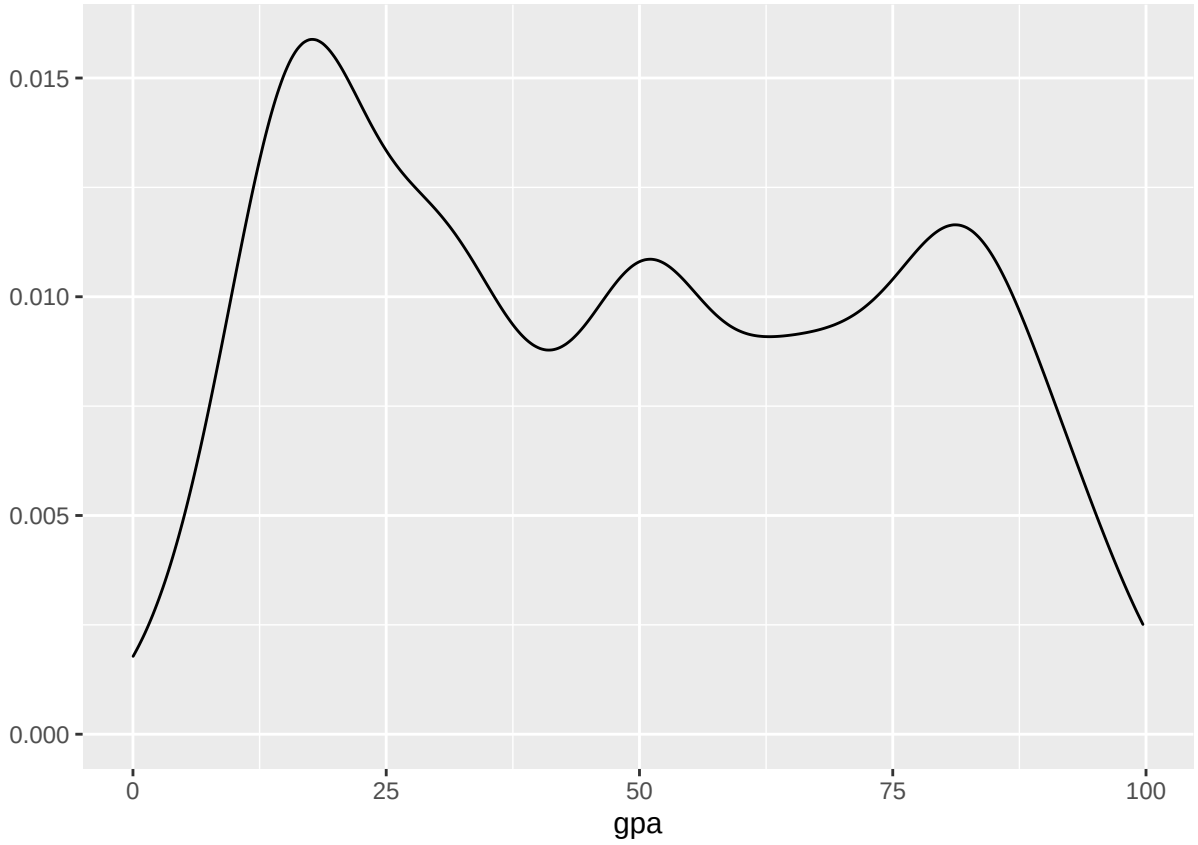
```
sd(grads$gpa)
```

```
## [1] 26.84777
```

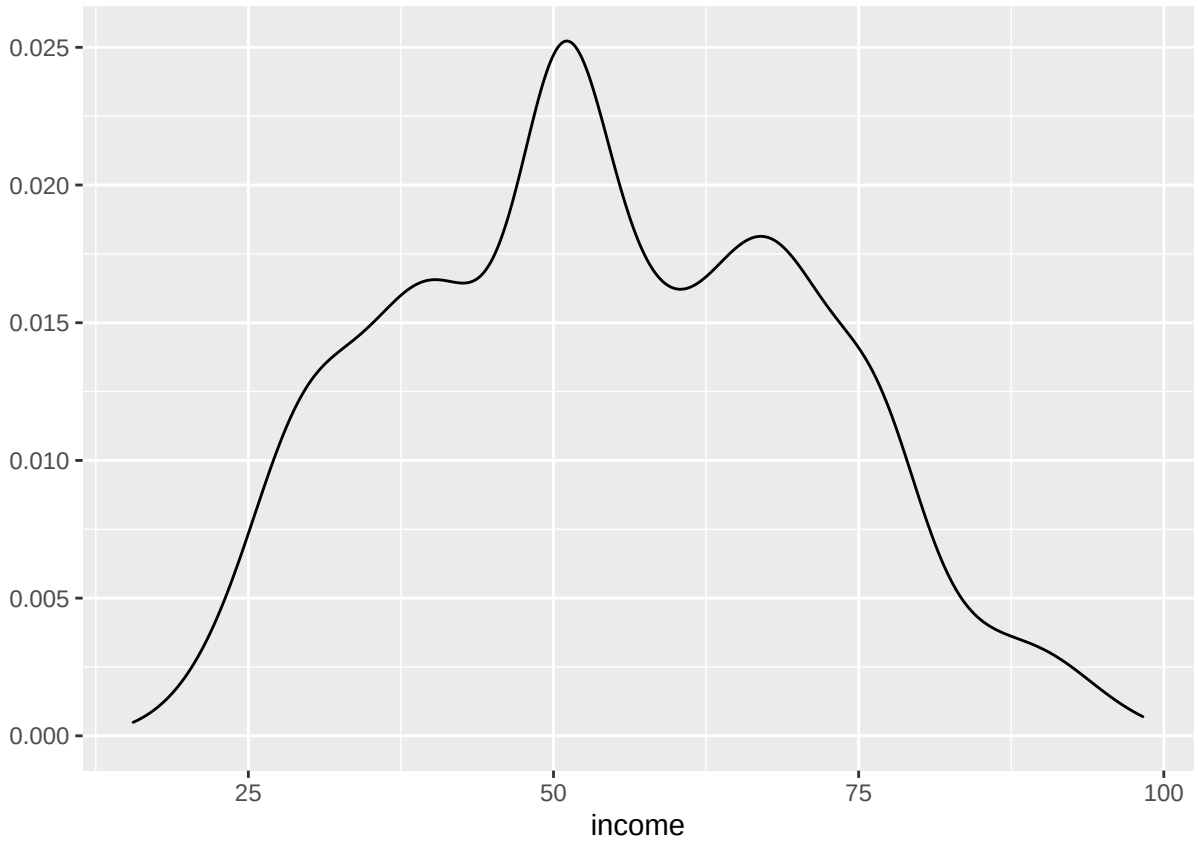
```
sd(grads$income)
```

```
## [1] 16.713
```

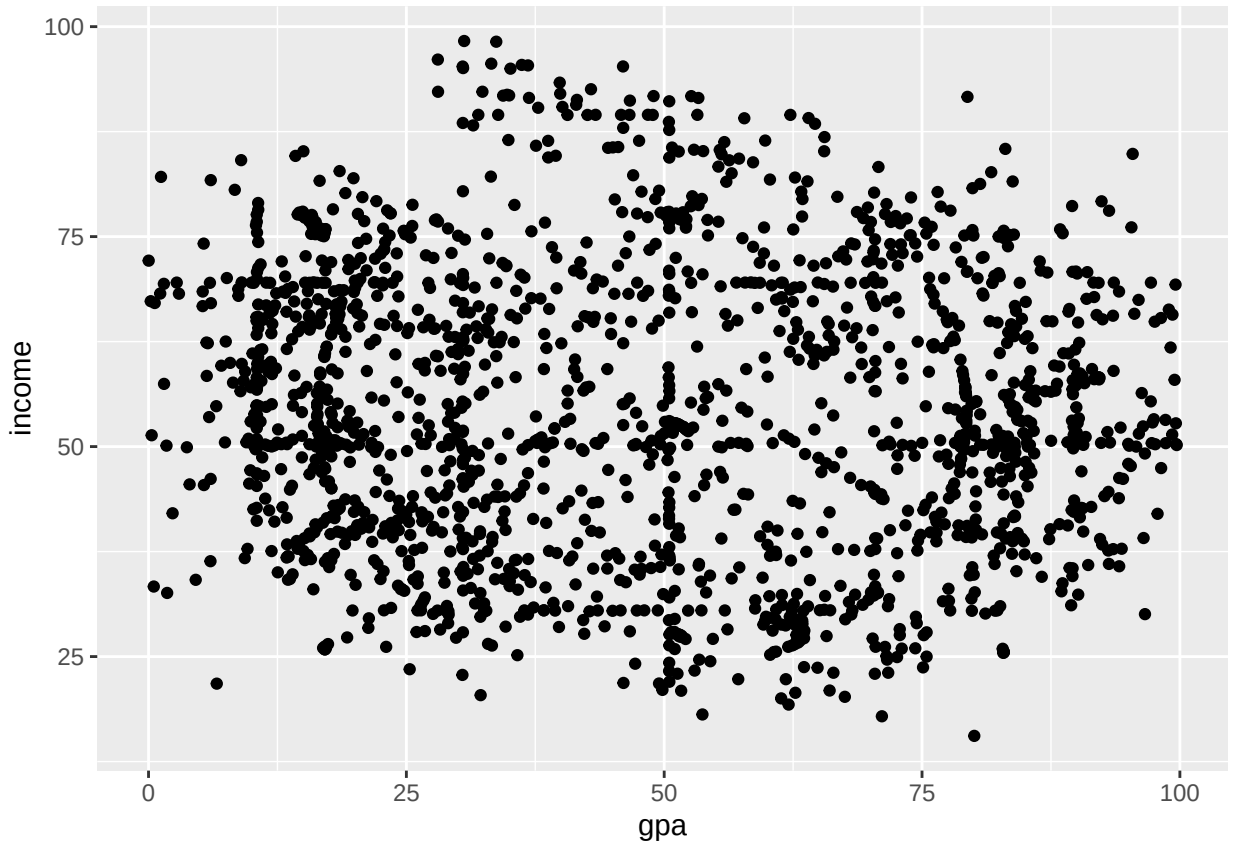
```
qplot(data = grads, x = gpa, geom = "density")
```




```
qplot(data = grads, x = income, geom = "density")
```




```
ggplot(data = grads, aes(x = gpa, y = income)) +
  geom_point()
```




I'll also calculate some summary statistics and visual comparisons within cohorts:

```
tapply(grads$income, grads$cohort, summary)
```

```
## $cohort_01
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   27.02   41.03   56.53   54.27   68.71   86.44
##
## $cohort_02
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   25.44   50.36   50.98   54.26   75.20   77.95
##
## $cohort_03
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   20.21   42.81   54.26   54.27   64.49   95.26
##
## $cohort_04
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   22.00   42.29   53.07   54.26   66.77   98.29
##
## $cohort_05
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   30.45   49.96   50.36   54.27   69.50   89.50
##
## $cohort_06
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   17.89   41.54   54.17   54.27   63.95   96.08
##
```

```
## $cohort_07
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   31.11   40.09   47.14   54.26   71.86   85.45
##
## $cohort_08
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   22.31   44.10   53.33   54.26   64.74   98.21
##
## $cohort_09
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   15.56   39.72   53.34   54.27   69.15   91.64
##
## $cohort_10
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   27.44   35.52   64.55   54.27   67.45   77.92
##
## $cohort_11
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   19.29   41.63   53.84   54.27   64.80   91.74
##
## $cohort_12
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   21.86   43.38   54.02   54.27   64.97   85.66
##
## $cohort_13
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   18.11   42.89   53.14   54.27   64.47   95.59
```

```
tapply(grads$income, grads$cohort, sd)
```

```
## cohort_01 cohort_02 cohort_03 cohort_04 cohort_05 cohort_06 cohort_07 cohort_08
##  16.76896  16.76774  16.76885  16.76590  16.76996  16.76670  16.76996  16.76514
## cohort_09 cohort_10 cohort_11 cohort_12 cohort_13
##  16.76982  16.77000  16.76924  16.76001  16.76676
```

```
tapply(grads$gpa, grads$cohort, summary)
```

```
## $cohort_01
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   14.37   20.37   50.11   47.84   63.55   92.21
##
## $cohort_02
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   15.77   17.11   51.30   47.84   82.88   94.25
##
## $cohort_03
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   5.646  24.756  45.292  47.831  70.856  99.580
##
## $cohort_04
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   10.46   30.48   50.47   47.83   70.35   90.46
##
## $cohort_05
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
```

```
##   2.735  22.753  47.114  47.837  65.845  99.695
##
## $cohort_06
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   14.91   22.92   32.50   47.84   75.94   87.15
##
## $cohort_07
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   4.578  23.471  39.876  47.840  73.610  97.838
##
## $cohort_08
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.949  25.288  46.026  47.832  68.526  99.487
##
## $cohort_09
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##  0.01512 24.62589 47.53527 47.83472 71.80315 97.47577
##
## $cohort_10
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.217  24.347  46.279  47.832  67.568  99.284
##
## $cohort_11
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   9.692  26.245  47.383  47.831  72.533  85.876
##
## $cohort_12
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   16.33   18.35   51.03   47.84   77.78   85.58
##
## $cohort_13
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.3039 27.8409 46.4013 47.8359 68.4394 99.6442
```

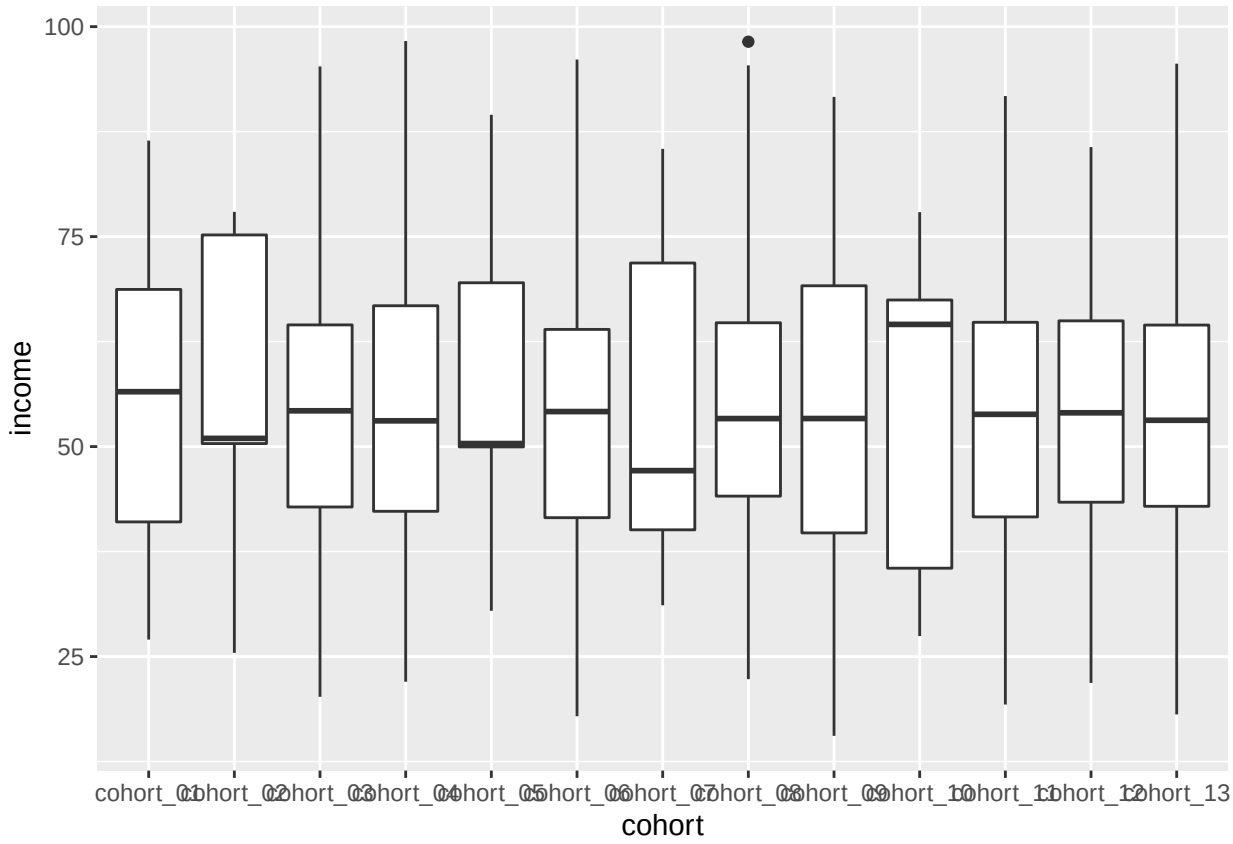
```
tapply(grads$gpa, grads$cohort, sd)
```

```
## cohort_01 cohort_02 cohort_03 cohort_04 cohort_05 cohort_06 cohort_07 cohort_08
##  26.93027  26.93019  26.93861  26.93988  26.93768  26.94000  26.93000  26.93540
## cohort_09 cohort_10 cohort_11 cohort_12 cohort_13
##  26.93974  26.93790  26.93573  26.93004  26.93610
```

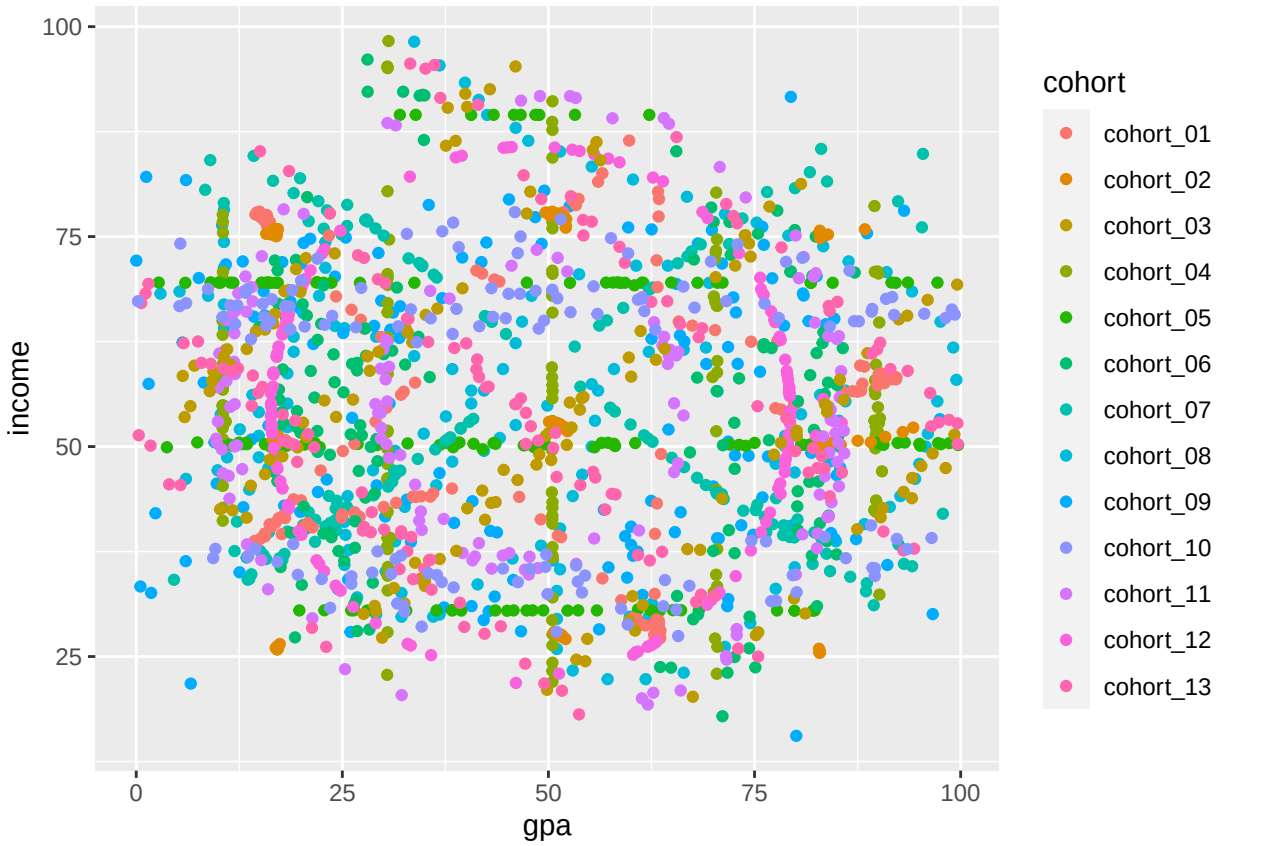
Huh. Those are remarkably similar values for the group means and the group standard deviations...

Onwards to plotting:

```
ggplot(data = grads, aes(x = cohort, y = income)) +
  geom_boxplot()
```

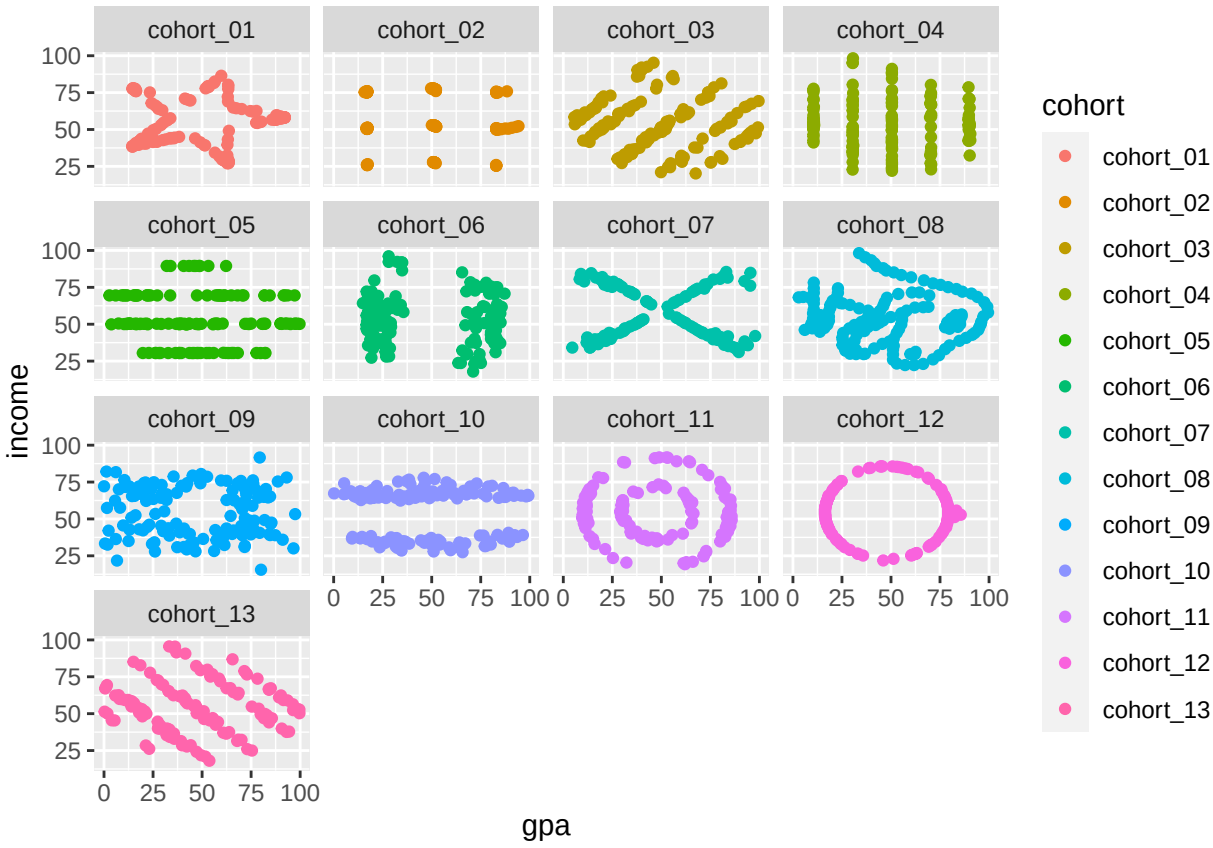



```
ggplot(data = grads, aes(x = gpa, y = income, color = cohort)) +
  geom_point()
```




Those plots are also a little strange. I know this is just a simulated analysis, but it still seems weird that overall it just looks like a big mass of random points, but when I add the colors by cohort, I can see there are some lines and other regularities within groups. I wonder what happens when I plot each scatter within cohorts?

```
ggplot(data = grads, aes(x = gpa, y = income, color = cohort)) +
  geom_point() +
  facet_wrap(vars(cohort))
```




Hmmm. That's... absurd (in particular, cohort 8 looks like a sideways dinosaur). At this point, if I were really working as a consultant on this project, I would write to the client and start asking some uncomfortable questions about data quality (who collected this data? how did it get recorded/stored/etc.? what quality controls were in place?). I would also feel obligated to tell them that there's just no way the data correspond to the variables they think are here. If you did that and the client was honest, they might tell you where the data actually came from.

In the event that you marched ahead with the analysis and are curious about what that could have looked like, I've provided some example code below. That said, *this is a situation where the assumptions and conditions necessary to identify ANOVA, t-tests, or regression are all pretty broken* because the data was generated programmatically in ways that undermine the kinds of interpretation you've been asked to make. The best response here (IMHO) is to abandon these kinds of analysis once you discover that there's something systematically weird going on. The statistical procedures will "work" in the sense that they will return a result, but because those results aren't even close to meaningful, any relationships you do observe in the data reflect something different than the sorts of relationships the statistical procedures were designed to identify.

```
summary(aov(income ~ cohort, data = grads)) # no global differences of means across groups
```

```
##               Df Sum Sq Mean Sq F value Pr(>F)
## cohort        12      0     0.0       0      1
## Residuals   1833 515354   281.1
```

```
summary(grads.model <- lm(income ~ gpa + cohort, data = grads)) # gpa percentile has a small, negative
```

```
##
## Call:
## lm(formula = income ~ gpa + cohort, data = grads)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -37.381 -13.259  -1.536  13.000  43.362
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      5.623e+01  1.567e+00  35.894  < 2e-16 ***
## gpa             -4.110e-02  1.451e-02  -2.832  0.00468 **
## cohortcohort_02 -7.026e-03  1.986e+00  -0.004  0.99718
## cohortcohort_03 -1.790e-03  1.986e+00  -0.001  0.99928
## cohortcohort_04 -6.281e-03  1.986e+00  -0.003  0.99748
## cohortcohort_05  2.481e-03  1.986e+00   0.001  0.99900
## cohortcohort_06  1.296e-03  1.986e+00   0.001  0.99948
## cohortcohort_07 -7.184e-03  1.986e+00  -0.004  0.99711
## cohortcohort_08 -4.368e-03  1.986e+00  -0.002  0.99825
## cohortcohort_09 -1.440e-03  1.986e+00  -0.001  0.99942
## cohortcohort_10 -7.512e-04  1.986e+00   0.000  0.99970
## cohortcohort_11  1.030e-03  1.986e+00   0.001  0.99959
## cohortcohort_12 -9.652e-05  1.986e+00   0.000  0.99996
## cohortcohort_13  3.578e-04  1.986e+00   0.000  0.99986
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.74 on 1832 degrees of freedom
## Multiple R-squared:  0.004359,   Adjusted R-squared:  -0.002707
## F-statistic: 0.6169 on 13 and 1832 DF,  p-value: 0.8419
```

```
confint(grads.model, "gpa") # 95% confidence interval
```

```
##           2.5 %      97.5 %
## gpa -0.06955974 -0.01263512
```

Note that the failure to reject the null of any association between district and income in the ANOVA does not provide conclusive evidence that the relationship between GPA and income does not vary by cohort. There were several things you might have done here. One is to calculate correlation coefficients within groups. Here's some tidyverse code that does that:

```
grads %>%
  group_by(cohort) %>%
  summarize(
    correlation = cor(income, gpa)
  )
```

```
## # A tibble: 13 x 2
##    cohort    correlation
##    <fct>           <dbl>
##  1 cohort_01     -0.0630
##  2 cohort_02     -0.0603
##  3 cohort_03     -0.0686
##  4 cohort_04     -0.0617
##  5 cohort_05     -0.0694
##  6 cohort_06     -0.0685
##  7 cohort_07     -0.0656
##  8 cohort_08     -0.0645
##  9 cohort_09     -0.0641
## 10 cohort_10     -0.0666
## 11 cohort_11     -0.0686
```

```
## 12 cohort_12     -0.0683
## 13 cohort_13     -0.0690
```

Because these correlation coefficients are nearly identical, I would likely end my analysis here and conclude that the correlation between gpa and income appears to be consistently small and negative. If you wanted to go further, you could theoretically calculate an interaction term in the model (by including `I(gpa*cohort)` in the model formula), but the analysis up to this point gives no indication that you'd be likely to find much of anything (and we haven't really talked about interactions yet).

# Part III: Trick or treating again

## Import and update data

```
## reminder that the "read_dta()" function requires the "haven" library

df <- read_dta(url("https://communitydata.science/~ads/teaching/2020/stats/data/week_07/Halloween2012-2

df <- df %>%
  mutate(
    obama = as.logical(obama),
    fruit = as.logical(fruit),
    year = as.factor(year),
    male = as.logical(male),
    age = age - mean(age, na.rm = T)
  )

df
```

```
## # A tibble: 1,223 x 7
##    obama fruit year     age male   neob treat_year
##    <lgl> <lgl> <fct>  <dbl> <lgl> <dbl>      <dbl>
##  1 FALSE FALSE 2014  -2.52  FALSE     1          4
##  2 FALSE TRUE  2014  -3.52  FALSE     1          4
##  3 FALSE FALSE 2014   0.480 TRUE      1          4
##  4 FALSE FALSE 2014  -3.52  TRUE      1          4
##  5 FALSE FALSE 2014  -1.52  FALSE     1          4
##  6 FALSE FALSE 2014   0.480 FALSE     1          4
##  7 FALSE FALSE 2014   1.48  TRUE      1          4
##  8 FALSE TRUE  2014  -3.52  FALSE     1          4
##  9 FALSE FALSE 2014  -0.520 TRUE      1          4
## 10 FALSE TRUE  2014  -1.52  FALSE     1          4
## # ... with 1,213 more rows
```

Let's fit and summarize the model:

```
f <- formula(fruit ~ obama + age + male + year)

fit <- glm(f, data = df, family = binomial("logit"))

summary(fit)
```

```
##
## Call:
## glm(formula = f, family = binomial("logit"), data = df)
##
```

```
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9079  -0.7853  -0.7319   1.4685   1.8134
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.2509167  0.2180417  -5.737 9.63e-09 ***
## obamaTRUE    0.2355872  0.1385821   1.700   0.0891 .
## age          0.0109508  0.0214673   0.510   0.6100
## maleTRUE    -0.1401293  0.1329172  -1.054   0.2918
## year2014     0.0002101  0.2215401   0.001   0.9992
## year2015     0.2600857  0.2107893   1.234   0.2173
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1374.9  on 1220  degrees of freedom
## Residual deviance: 1367.2  on 1215  degrees of freedom
##   (2 observations deleted due to missingness)
## AIC: 1379.2
##
## Number of Fisher Scoring iterations: 4
```

Interesting. Looks like adjusting for these other variables in a regression setting can impact the results.

Onwards to generating more interpretable results:

```
## Odds ratios (exponentiated log-odds!)
exp(coef(fit))
```

```
## (Intercept)   obamaTRUE         age    maleTRUE    year2014    year2015
##   0.2862423   1.2656518   1.0110110   0.8692458   1.0002101   1.2970413
```

```
exp(confint(fit))
```

```
##                 2.5 %    97.5 %
## (Intercept) 0.1846712 0.4348622
## obamaTRUE   0.9635277 1.6593989
## age         0.9691402 1.0543000
## maleTRUE    0.6697587 1.1280707
## year2014    0.6518923 1.5564945
## year2015    0.8649706 1.9799543
```

```
## model-predicted probabilities for prototypical observations:
fake.kids <- data.frame(
  obama = rep(c(FALSE, TRUE), 2),
  year = factor(rep(c("2015", "2012"), 2)),
  age = rep(c(9, 7), 2),
  male = rep(c(FALSE, TRUE), 2)
)

fake.kids.pred <- cbind(fake.kids, pred.prob = predict(fit, fake.kids, type = "response"))

fake.kids.pred
```

```
##   obama year age  male pred.prob
```

```
## 1 FALSE 2015   9 FALSE 0.2906409
## 2  TRUE 2012   7  TRUE 0.2537326
## 3 FALSE 2015   9 FALSE 0.2906409
## 4  TRUE 2012   7  TRUE 0.2537326
```

Note that this UCLA logit regression tutorial also contains example code to help extract standard errors and confidence intervals around these predicted probabilities. You were not asked to produce them here, but if you'd like an example here you go (I can try to clarify in class):

```
fake.kids.more.pred <- cbind(
  fake.kids,
  predict(fit, fake.kids, type = "link", se = TRUE)
)

within(fake.kids.more.pred, {
  pred.prob <- plogis(fit)
  lower <- plogis(fit - (1.96 * se.fit))
  upper <- plogis(fit + (1.96 * se.fit))
})
```

```
##   obama year age  male        fit    se.fit residual.scale     upper     lower
## 1 FALSE 2015   9 FALSE -0.8922736 0.2243091              1 0.3887361 0.2088421
## 2  TRUE 2012   7  TRUE -1.0788031 0.2594135              1 0.3611555 0.1697706
## 3 FALSE 2015   9 FALSE -0.8922736 0.2243091              1 0.3887361 0.2088421
## 4  TRUE 2012   7  TRUE -1.0788031 0.2594135              1 0.3611555 0.1697706
##   pred.prob
## 1 0.2906409
## 2 0.2537326
## 3 0.2906409
## 4 0.2537326
```

## Sub-group analysis

```
f2 <- formula(fruit ~ obama + age + male)

summary(glm(f2, data = df[df$year == "2012", ], family = binomial("logit")))
```

```
##
## Call:
## glm(formula = f2, family = binomial("logit"), data = df[df$year ==
##     "2012", ])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0041  -0.7564  -0.6804  -0.5434   1.9929
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.53780    0.37711  -4.078 4.55e-05 ***
## obamaTRUE    0.32568    0.38680   0.842    0.400
## age          0.08532    0.06699   1.273    0.203
## maleTRUE     0.32220    0.38802   0.830    0.406
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 177.56  on 163  degrees of freedom
## Residual deviance: 174.90  on 160  degrees of freedom
##   (1 observation deleted due to missingness)
## AIC: 182.9
##
## Number of Fisher Scoring iterations: 4
```

```
summary(glm(f2, data = df[df$year == "2014", ], family = binomial("logit")))
```

```
##
## Call:
## glm(formula = f2, family = binomial("logit"), data = df[df$year ==
##     "2014", ])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.8016  -0.7373  -0.6846  -0.6594   1.8071
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.13351    0.19798  -5.725 1.03e-08 ***
## obamaTRUE    0.07348    0.24158   0.304    0.761
## age          0.01198    0.03673   0.326    0.744
## maleTRUE    -0.23980    0.23499  -1.020    0.308
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 450.11  on 421  degrees of freedom
## Residual deviance: 448.85  on 418  degrees of freedom
## AIC: 456.85
##
## Number of Fisher Scoring iterations: 4
```

```
summary(glm(f2, data = df[df$year == "2015", ], family = binomial("logit")))
```

```
##
## Call:
## glm(formula = f2, family = binomial("logit"), data = df[df$year ==
##     "2015", ])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9067  -0.7931  -0.7338   1.4761   1.7012
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.9971079  0.1383417  -7.208  5.7e-13 ***
## obamaTRUE    0.3170892  0.1898404   1.670   0.0949 .
## age          0.0004615  0.0290552   0.016   0.9873
## maleTRUE    -0.1791721  0.1791828  -1.000   0.3173
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 743.8  on 634  degrees of freedom
## Residual deviance: 740.0  on 631  degrees of freedom
##   (1 observation deleted due to missingness)
## AIC: 748
##
## Number of Fisher Scoring iterations: 4
```

Interesting. The treatment effect seems to emerge overwhelmingly within a single year of the data.