Part 1

Recreate the follwing in R Markdown, with your code included:

Gender Bias in the Movie Biz

In 1985, cartoonist Alison Bechdel (http://dykestowatchoutfor.com/) proposed “The Rule.” To pass, a movie has to sastisfy three basic requirements:

1. It has to have at least two women in it,
2. The two women have to talk to each other, and
3. They have to talk to each other about something besides a man.

“The Rule” (see a copy of the original comic strip here (http://www.npr.org/templates/story/story.php?storyId=94202522)) is commonly refered to as the Bechdel test. It’s a seemingly low bar, and it’s surprising how many films **fail** the test.

In 2014, FiveThirtyEight (http://fivethirtyeight.com/) analyzed 1,615 films between 1990 and 2013 to explore the financial effect of a these films’ portrayal of women in an article titled The Dollar-And-Cents Case Against Hollywood’s Exclusion of Women (http://fivethirtyeight.com/features/the-dollar-and-cents-case-against-hollywoods-exclusion-of-women/). Their analysis relied on data sets from BechdelTest.com (http://bechdeltest.com/) and The-Numbers.com (http://www.the-numbers.com/). They concluded that the “the median bidget of movies that passed the test - those that featured a conversation between two women about something other than a man — was substantially lower than the median budget of all films in the sample.” Overall, movies that passed the test may have a better financial return than those that don’t.

The data

The data set used by FiveThirtyEight is available for download here (https://github.com/fivethirtyeight/data/blob/master/bechdel/movies.csv) from FiveThirtyEight’s GitHub data page (https://github.com/fivethirtyeight/data). Download and read in the data set, and look at the first few rows of the dataframe:

```
##   year      imdb            title            test clean_test binary
## 1 2013 tt1711425    21 &amp;amp; Over          notalk     notalk   FAIL
## 2 2012 tt1343727         Dredd 3D     ok-disagree         ok   PASS
## 3 2013 tt2024544 12 Years a Slave notalk-disagree     notalk   FAIL
##     budget domgross  intgross     code budget_2013. domgross_2013.
## 1 13000000 25682380  42195766 2013FAIL     13000000       25682380
## 2 45000000 13414714  40868994 2012PASS     45658735       13611086
## 3 20000000 53107035 158607035 2013FAIL     20000000       53107035
##   intgross_2013. period.code decade.code
## 1       42195766           1           1
## 2       41467257           1           1
## 3      158607035           1           1
```

Here are the last few rows:

```
##      year      imdb                                    title         test
## 1792 1971 tt0067116                  The French Connection       notalk
## 1793 1971 tt0067992 Willy Wonka &amp;amp; the Chocolate Factory men-disagree
## 1794 1970 tt0065466          Beyond the Valley of the Dolls           ok
##      clean_test binary  budget domgross intgross     code budget_2013.
## 1792     notalk   FAIL 2200000 41158757 41158757 1971FAIL     12659931
## 1793        men   FAIL 3000000  4000000  4000000 1971FAIL     17263543
## 1794         ok   PASS 1000000  9000000  9000000 1970PASS      5997631
##      domgross_2013. intgross_2013. period.code decade.code
## 1792      236848653      236848653          NA          NA
## 1793       23018057       23018057          NA          NA
## 1794       53978683       53978683          NA          NA
```

Right now, the data set includes movies from 1970 to 2013. We want to make sure we only include movies that came out *after* 1990. Here are the new last few rows in the `year` and `title` columns:

```
##
## 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
```

```
##      year                   title
## 1613 1990 The Hunt for Red October
## 1614 1990             Total Recall
## 1615 1990                  Tremors
```

We can probably clean up a few of the column names to make them easier to work with. Here are the current column names:

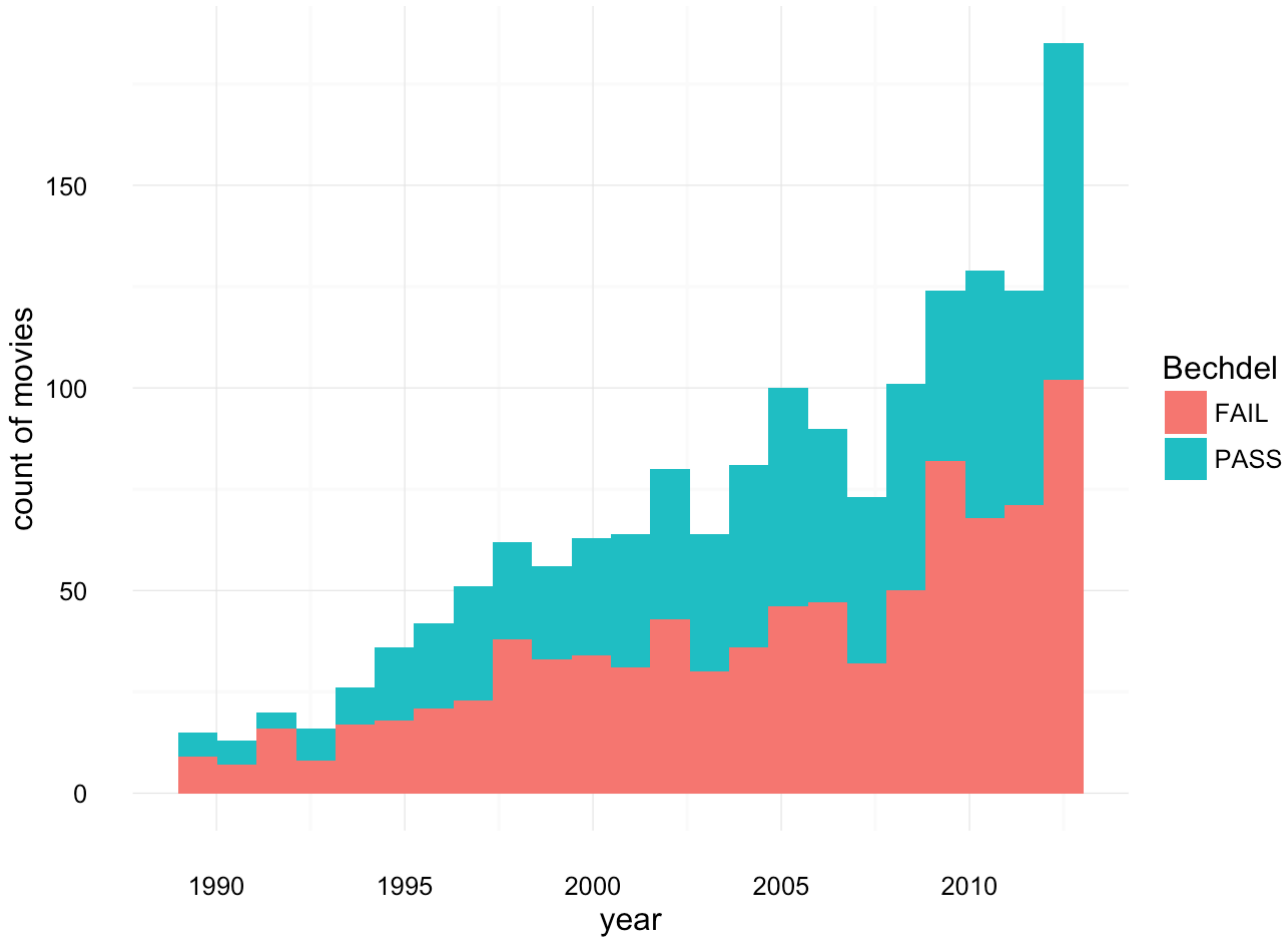
```
##  [1] "year"           "imdb"           "title"          "test"
##  [5] "clean_test"     "binary"         "budget"         "domgross"
##  [9] "intgross"       "code"           "budget_2013."   "domgross_2013."
## [13] "intgross_2013." "period.code"    "decade.code"
```

It might be useful to rename “binary” to “Bechdel”, and to remove the periods `.` from “budget_2013.”, “domgross_2013.”, and “intgross_2013.”:

```
##  [1] "year"          "imdb"          "title"         "test"
##  [5] "clean_test"    "Bechdel"       "budget"        "domgross"
##  [9] "intgross"      "code"          "budget_2013"   "domgross_2013"
## [13] "intgross_2013" "period.code"   "decade.code"
```

“The Rule”

It looks like `binary` tells us whether a movie passed or failed. Here's how the count of movies that have failed and passed the test has changed over time:



`clean_test` offers a finer detail of the Bechdel Test than Fail vs. Pass. Here are the different levels of this variable:

```
## [1] notalk  ok      men     nowomen dubious
## Levels: dubious men notalk nowomen ok
```

Based on FiveThirtyEight's article (http://fivethirtyeight.com/features/the-dollar-and-cents-case-against-hollywoods-exclusion-of-women/), it seems like these levels correspond to the following categories:

clean_test	Category_Description	Bechdel_Test
nowomen	Fewer than two women	Fail
notalk	Women don't talk to each other	Fail
men	Women only talk about men	Fail
dubious	Dubious	Pass
ok	Passes Bechdel Test	Pass

We can check out the median budget in 2013 dollars for movies fitting into each of these categories. First, create a dataframe with the median value of `budget_2013` for each movie (in ascending order) using `dplyr` 's `group_by()`, `summarize()`, and `arrange()` functions. Change the dollar amount so that its units are in

millions of dollars.

```
## # A tibble: 5 × 2
##   clean_test median_budget_2013
##       <fctr>              <dbl>
## 1         ok           31.07072
## 2    dubious           35.79099
## 3        men           39.73769
## 4    nowomen           43.37307
## 5     notalk           56.57008
```

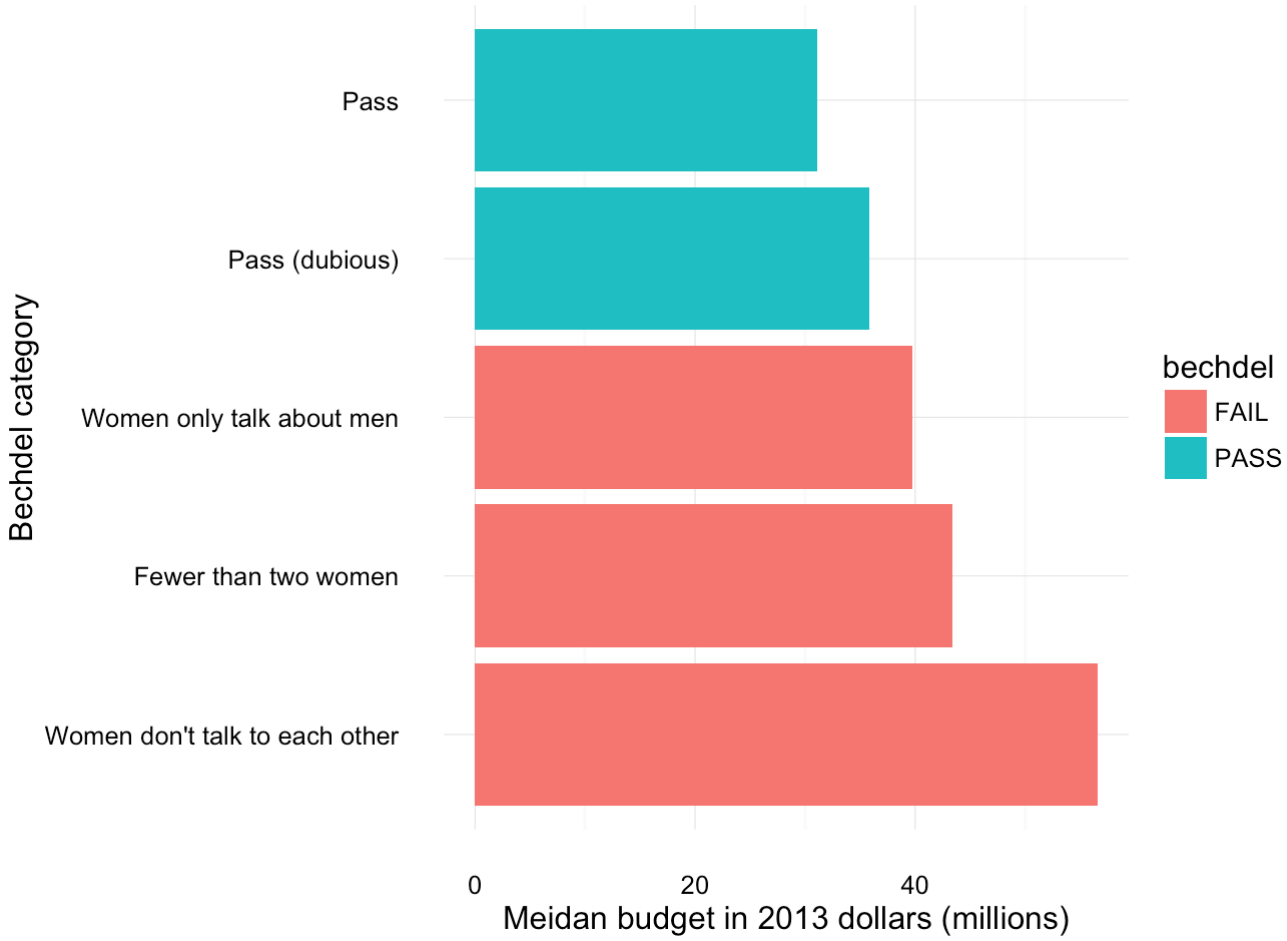
Next, for the purposes of later plotting, reorder the levels of `clean_test` so that they are in *descending* order, and rename them to more verbose descriptions: “Women don’t talk to each other”, “Fewer than two women”, “Women only talk about men”, “Pass (dubious)”, and “Pass”, for “notalk”, “nowomen”, “men”, “dubious”, and “ok”, respectively.

```
## # A tibble: 5 × 2
##                       clean_test median_budget_2013
##                           <fctr>              <dbl>
## 1                           Pass           31.07072
## 2                Pass (dubious)           35.79099
## 3      Women only talk about men           39.73769
## 4           Fewer than two women           43.37307
## 5 Women don't talk to each other           56.57008
```

Add a column of corresponding Bechdel test results (one way to do this is to create a vector of “PASS” and “FAIL” values, and then `cbind` it to the dataframe):

```
##                       clean_test median_budget_2013 bechdel
## 1                           Pass           31.07072    PASS
## 2                Pass (dubious)           35.79099    PASS
## 3      Women only talk about men           39.73769    FAIL
## 4           Fewer than two women           43.37307    FAIL
## 5 Women don't talk to each other           56.57008    FAIL
```

And finally, following FiveThirtyEight’s (http://fivethirtyeight.com/features/the-dollar-and-cents-case-against-hollywoods-exclusion-of-women/) lead, plot this dataframe as a bar graph:



We can see that past movies that have failed the Bechdel test had higher median budgets. What about how much money these movies made? In FiveThirtyEight's analysis, they focused on "Return on investment", which involved dividing movie profits (`domgross_2013` , for example) by movie budgets. For simplicity's sake, let's compare domestic gross for movies that passed and failed the test. Again, we should `mutate domgross_2013` so that it is in units of millions of dollars.

First, change `domgross_2013` to numeric class. Currently, it's saved as a factor:

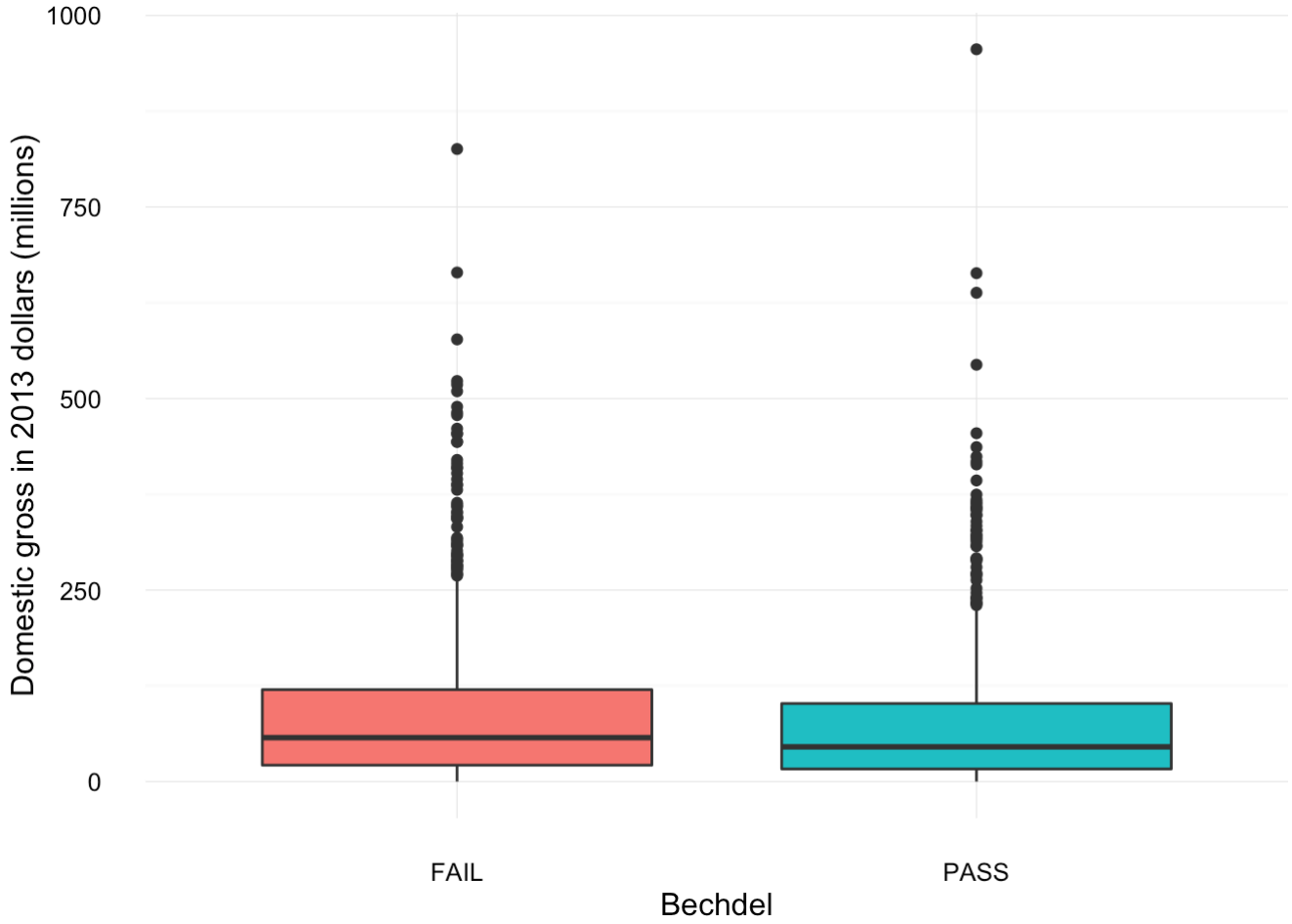
```
## [1] "factor"
```

First convert the column class to character, and then numeric.

```
## Warning: NAs introduced by coercion
```

Here's a boxplot comparing domestic gross for movies that failed and passed the Bechdel test:

```
## Warning: Removed 15 rows containing non-finite values (stat_boxplot).
```



With a lot of outliers, it looks like the mean domestic gross for movies that failed is *slightly* higher than that for movies that passed.

Let's find out what the highest grossing movies in "Pass" and "Fail" are. For failed, it looks like the higest grossing movie in this data set that failed the Bechdel test earned $825.7 million (in 2013 dollars):

```
## [1] 825.7072
```

And its title is "Avatar"!

```
## [1] Avatar
## 1768 Levels: (500) Days of Summer [Rec] ... Zwartboek
```

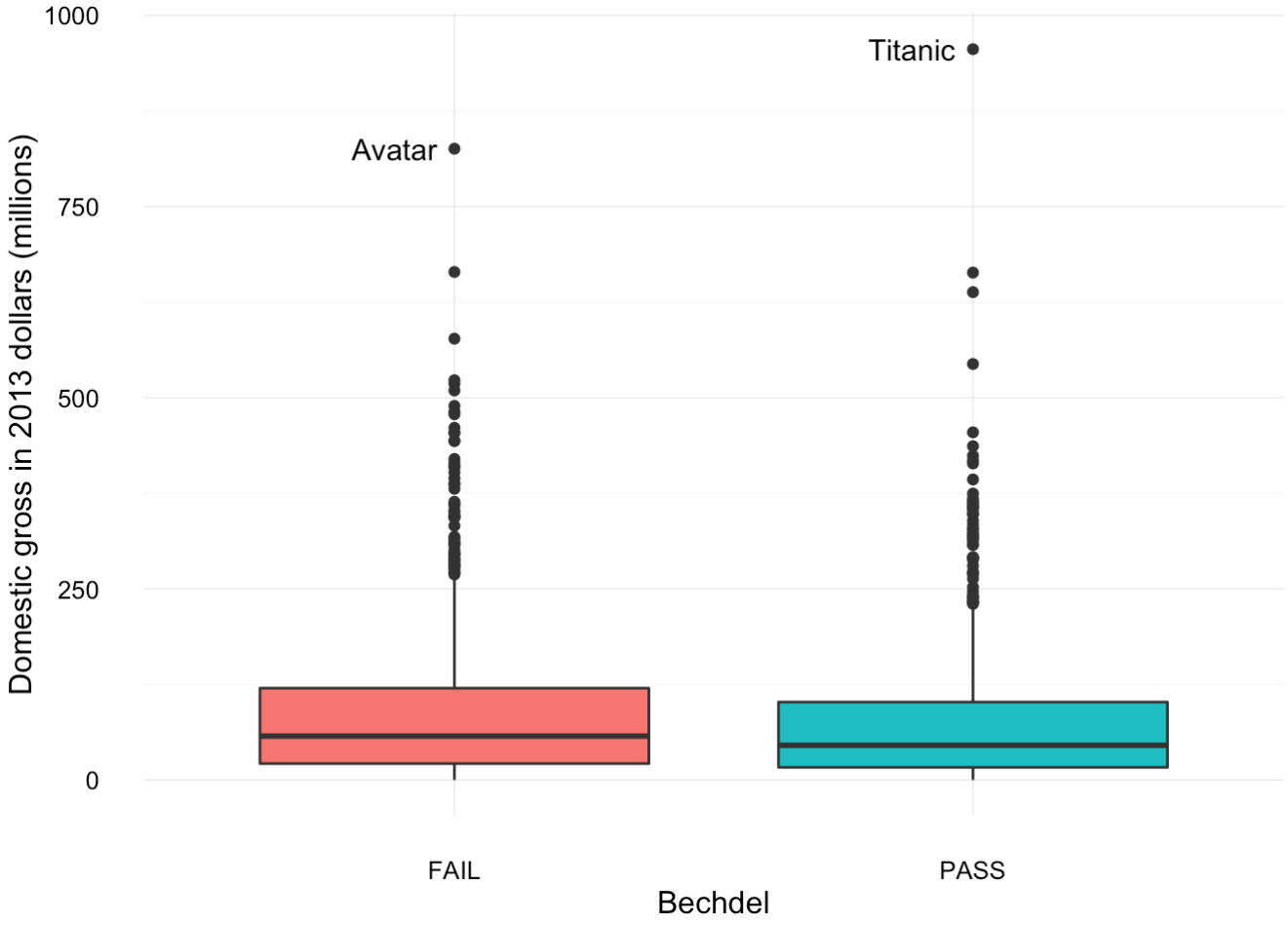
For passed, it looks like "Titanic" made $955.9 million (in 2013 dollars).

```
## [1] 955.8904
```

```
## [1] Titanic
## 1768 Levels: (500) Days of Summer [Rec] ... Zwartboek
```

To wrap up, add both of these titles as interesting labels to the plot. (Note: to match the lables shown here, use `size = 4` and `hjust = 1.2` in your `geom_text()` addition.)

```
## Warning: Removed 15 rows containing non-finite values (stat_boxplot).
```


1000
750
500
250
0
Domestic gross in 2013 dollars (millions)
Titanic
Avatar
FAIL
PASS
Bechdel