Lab 3: Data Manipulation

Derek Hansen (credit to Brian Manzo)

Wednesday, September 22, 2020 1:00PM-2:30PM

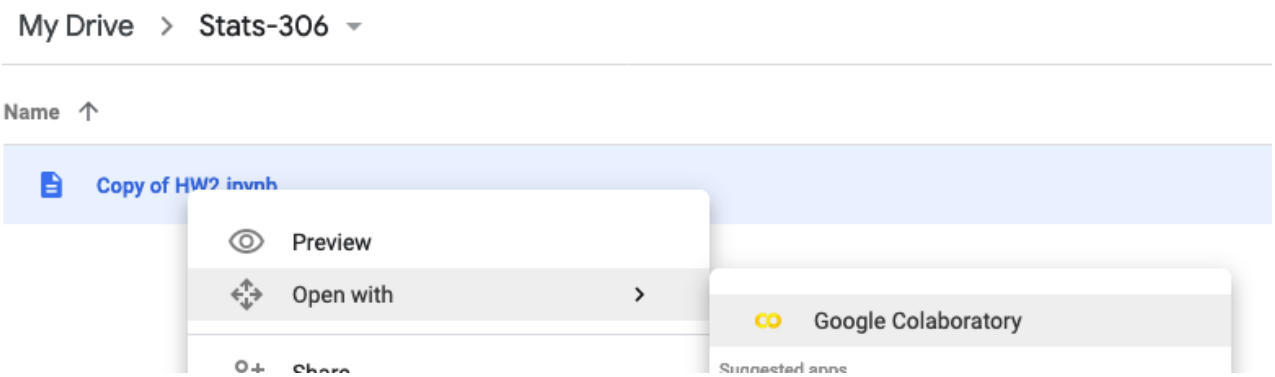
Preliminaries

1. Piazza reminders
    - When asking a homework question, gather all relevant information to your problem to make it *reproduceable*
        - What operating system are you using (Windows/Mac)
        - What version of R are you using?
        - What versions of packages are you using? (run sessionInfo() after loading packages to see)
        - What code did you run before running into your problem?
        - What does the output say?
        - Take a screenshot or picture for best results
2. I've finished grading HW0 and HW1; grades should be up shortly.
    - Rubric and answer sheet posted
    - Not feasible to leave comments on Jupyter
    - Come to office hours if you want to discuss your homework
3. Google Colab demonstration

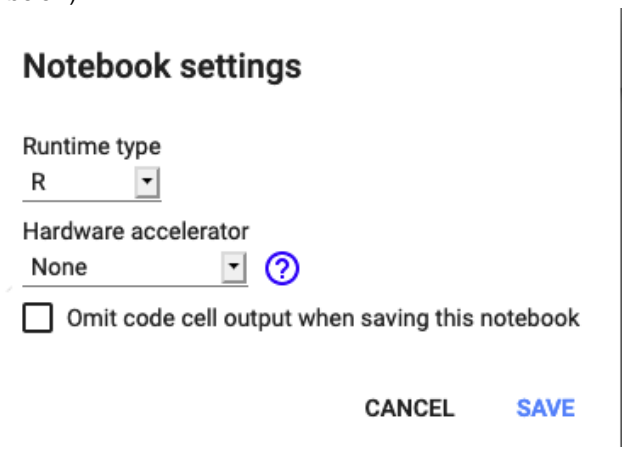
Google CoLab

Here are some instructions for getting started with Colab.

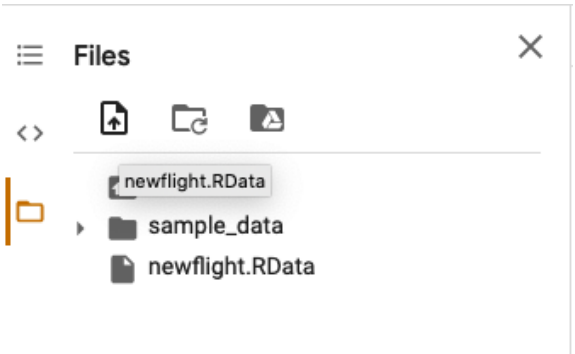
1. Download the homework/lecture from canvas (.ipynb file).
2. Open Google colab (https://colab.research.google.com/notebooks/ (https://colab.research.google.com/notebooks/)) and choose the "upload" option. Choose the file you've downloaded from Canvas. You can also just upload the notebook to your google drive, and then choose open with Google Colaboratory (as I show in the screenshot). If you want to make a new notebook from scratch, you can use this link (https://colab.research.google.com/#create=true&language=r (https://colab.research.google.com/#create=true&language=r))



   A. To make sure the notebook is running R, and not Python, you can check Runtime --> choose runtime type --> make sure it says R (like the picture below).



3. Use the notebook as you would in Jupyter. If you need to use an external dataset (i.e., newflight), you can upload it to the notebook by choosing the folder icon on the side of your screen and then clicking the "upload" icon on the left. The data won't stay there across sessions, so the only annoying thing here is you have to reupload the data each time you work on the assignment (hopefully this isn't too burdensome).



4. I encountered an error when making certain kinds of ggplots. If you also get an error, you can try adding this snippet of code after you load tidyverse.

require(devtools) install_version("farver", version = "2.0.2", repos = "http://cran.us.r-project.org") library(farver)

2013 Men's French Open (Tennis)

In [1]:
```
library(tidyverse) # load tidyverse
```

```
── Attaching packages ─────────────────────────────────────── tidyverse 1.3.0 ──

✔ ggplot2 3.3.2     ✔ purrr   0.3.4
✔ tibble  3.0.3     ✔ dplyr   1.0.2
✔ tidyr   1.1.2     ✔ stringr 1.4.0
✔ readr   1.3.1     ✔ forcats 0.4.0

── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
```

In [2]:
```
tennis_data = read_csv('FrenchOpen-men-2013.csv')
```

```
Parsed with column specification:
cols(
  .default = col_double(),
  Player1 = col_character(),
  Player2 = col_character()
)

See spec(...) for full column specifications.
```

You can download data directly from websites if you pass the URL. This is useful for downloading from Github, and especially useful for Google CoLab.

In [3]:
```
tennis_data = read_csv('https://raw.githubusercontent.com/bmanzo/stats306_labs/master/lab03/FrenchOpen-men-2013.csv')
```

```
Parsed with column specification:
cols(
  .default = col_double(),
  Player1 = col_character(),
  Player2 = col_character()
)

See spec(...) for full column specifications.
```

In [4]:
```
head(tennis_data)
```

Player1	Player2	Round	Result	FNL.1	FNL.2	FSP.1	FSW.1	SSP.1	SSW.1	...	BPC.2	BPW.2	NPA.2	NP
Pablo Carreno-Busta	Roger Federer	1	0	0	3	62	27	38	11	...	7	7	14	
Somdev Devvarman	Daniel Munoz-De La Nava	1	1	3	0	62	54	38	22	...	1	16	22	
Tobias Kamke	Paolo Lorenzi	1	1	3	2	62	53	38	15	...	10	18	19	
Julien Benneteau	Ricardas Berankis	1	1	3	1	72	87	28	19	...	4	13	33	
Lukas Lacko	Sam Querrey	1	0	0	3	52	31	48	22	...	4	7	12	
Jan Hajek	Denis Kudla	1	1	3	1	70	58	30	18	...	1	7	6	

https://archive.ics.uci.edu/ml/datasets/Tennis+Major+Tournament+Match+Statistics# (https://archive.ics.uci.edu/ml/datasets/Tennis+Major+Tournament+Match+Statistics#)

`dplyr` functions

The `filter` function is used to retrieve a subset of the full dataset

Roger Federer is a very famous tennis player. Let's use `filter` to find all the matches in which he played in the 2013 French Open.

In [5]:
```
(federer <- tennis_data %>%
            filter(Player1=='Roger Federer' | Player2=='Roger Federer'))
```

Player1	Player2	Round	Result	FNL.1	FNL.2	FSP.1	FSW.1	SSP.1	SSW.1	...	BPC.2	BPW.2	NPA.2	NP
Pablo Carreno-Busta	Roger Federer	1	0	0	3	62	27	38	11	...	7	7	14	
Somdev Devvarman	Roger Federer	2	0	0	3	61	19	39	16	...	7	14	19	
Julien Benneteau	Roger Federer	3	0	0	3	82	41	18	8	...	4	4	8	
Gilles Simon	Roger Federer	4	0	2	3	61	65	39	28	...	6	14	25	
Jo-Wilfried Tsonga	Roger Federer	5	1	3	0	75	46	25	10	...	2	3	15	

If you want to assign as well as print the variable, enclose the command in parentheses.

The above table is useful, but we don't need all of the columns. We can use `select` to only show a subset of the columns. Create a new table, `fed_select`, which only shows the fields `Player1`, `Player2`, `Round`, and `Result`.

In [6]:
```
fed_select <- tennis_data %>%
     filter(Player1=='Roger Federer' | Player2=='Roger Federer') %>%
     select(Player1:Result)
```

In [7]:
```
fed_select
```

Player1	Player2	Round	Result
Pablo Carreno-Busta	Roger Federer	1	0
Somdev Devvarman	Roger Federer	2	0
Julien Benneteau	Roger Federer	3	0
Gilles Simon	Roger Federer	4	0
Jo-Wilfried Tsonga	Roger Federer	5	1

In [8]:
```
print(fed_select)
```

```
# A tibble: 5 x 4
  Player1             Player2       Round Result
  <chr>               <chr>         <dbl>  <dbl>
1 Pablo Carreno-Busta Roger Federer     1      0
2 Somdev Devvarman    Roger Federer     2      0
3 Julien Benneteau    Roger Federer     3      0
4 Gilles Simon        Roger Federer     4      0
5 Jo-Wilfried Tsonga  Roger Federer     5      1
```

We can use functions such as `between` or the `%in%` operator.

In [9]:
```
top_three = tennis_data %>%
                filter(Player1 %in% c('Roger Federer', 'Novak Djokovic', 'Rafael N
adal') | Player2 %in% c('Roger Federer', 'Novak Djokovic', 'Rafael Nadal'))
```

In [10]:
```
middle_round = tennis_data %>%
                filter(between(Round, 3, 5))
```

Suppose we are interested in the later rounds of the tournament. We can use the `arrange` function to order rows instead of filtering for a subset of them.

In [11]:
```
tennis_data %>% arrange(desc(Round))
```

Player1	Player2	Round	Result	FNL.1	FNL.2	FSP.1	FSW.1	SSP.1	SSW.1	...	BPC.2	BPW.2	NPA.2
Rafael Nadal	David Ferrer	7	1	3	0	70	43	30	11	...	3	12	10
David Ferrer	Jo-Wilfried Tsonga	6	1	3	0	60	35	40	23	...	2	5	7
Novak Djokovic	Rafael Nadal	6	0	2	3	67	76	33	30	...	8	16	15
Jo-Wilfried Tsonga	Roger Federer	5	1	3	0	75	46	25	10	...	2	3	15
Tommy Robredo	David Ferrer	5	0	0	3	59	22	41	9	...	7	12	13
Rafael Nadal	Stanislas Wawrinka	5	1	3	0	75	40	25	11	...	1	5	16
Novak Djokovic	Tommy Haas	5	1	3	0	64	41	36	22	...	2	2	2
Gilles Simon	Roger Federer	4	0	2	3	61	65	39	28	...	6	14	25
Jo-Wilfried Tsonga	Viktor Troicki	4	1	3	0	77	45	23	15	...	0	3	19
Kevin Anderson	David Ferrer	4	0	0	3	71	34	29	7	...	6	17	9
Tommy Robredo	Nicolas Almagro	4	1	3	2	63	73	37	31	...	6	13	19
Stanislas Wawrinka	Richard Gasquet	4	1	3	2	63	81	37	41	...	2	11	37
Rafael Nadal	Kei Nishikori	4	1	3	0	69	39	31	16	...	0	4	8
Tommy Haas	Mikhail Youzhny	4	1	3	0	61	31	39	14	...	2	6	13
Novak Djokovic	Philipp Kohlschreiber	4	1	3	1	74	71	26	23	...	2	13	17
Julien Benneteau	Roger Federer	3	0	0	3	82	41	18	8	...	4	4	8
Gilles Simon	Sam Querrey	3	1	3	2	60	52	40	32	...	7	14	21
Viktor Troicki	Marin Cilic	3	1	3	0	65	61	35	20	...	2	9	20
Jo-Wilfried Tsonga	Jeremy Chardy	3	1	3	0	67	40	33	17	...	0	2	15
Feliciano Lopez	David Ferrer	3	0	0	3	63	37	37	14	...	7	19	14
Milos Raonic	Kevin Anderson	3	0	0	3	57	49	43	29	...	3	12	10
Andreas Seppi	Nicolas Almagro	3	0	0	3	51	34	49	18	...	5	13	12
Gael Monfils	Tommy Robredo	3	0	2	3	68	73	32	27	...	6	17	28
Nikolay Davydenko	Richard Gasquet	3	0	0	3	76	46	24	9	...	5	10	17

Player1	Player2	Round	Result	FNL.1	FNL.2	FSP.1	FSW.1	SSP.1	SSW.1	...	BPC.2	BPW.2	NPA.2
Stanislas Wawrinka	Jerzy Janowicz	3	1	3	1	58	54	42	31	...	1	5	19
Benoit Paire	Kei Nishikori	3	0	1	3	43	40	57	36	...	8	19	14
Rafael Nadal	Fabio Fognini	3	1	3	0	78	52	22	14	...	3	11	17
Mikhail Youzhny	Janko Tipsarevic	3	1	3	0	64	40	36	21	...	0	3	10
Tommy Haas	John Isner	3	0	2	3	63	99	37	52	...	2	13	52
Victor Hanescu	Philipp Kohlschreiber	3	0	0	3	73	31	27	8	...	6	9	19
...	...	...	...	...	...	...	...	...	...		...	...	...
Michal Przysiezny	Rhyne Williams	1	1	3	1	67	75	33	27	...	1	10	6
Florent Serra	Nikolay Davydenko	1	0	0	3	64	35	36	13	...	5	7	5
Florian Mayer	Denis Istomin	1	0	1	2	59	40	41	15	...	5	11	17
Albert Ramos	Jerzy Janowicz	1	0	0	3	61	41	39	25	...	3	7	19
Kenny De Schepper	Robin Haase	1	0	1	3	64	61	36	21	...	3	8	17
Vasek Pospisil	Horacio Zeballos	1	0	2	2	56	67	44	34	...	1	10	15
Stanislas Wawrinka	Thiemo De Bakker	1	1	3	1	69	74	31	24	...	1	4	20
Jesse Levine	Kei Nishikori	1	0	0	3	52	16	48	9	...	8	11	7
Grega Zemlja	Santiago Giraldo	1	1	3	0	52	32	48	21	...	0	3	1
Lukasz Kubot	Maxime Teixeira	1	1	2	1	64	30	36	12	...	5	8	11
Benoit Paire	Marcos Baghdatis	1	1	2	1	46	40	54	29	...	2	14	13
Andreas Beck	Fabio Fognini	1	0	0	3	55	37	45	25	...	5	13	9
Pere Riba	Lukas Rosol	1	0	0	3	60	38	40	14	...	4	7	9
Martin Klizan	Michael Russell	1	1	2	1	49	29	51	22	...	3	12	10
Rafael Nadal	Daniel Brands	1	1	3	1	84	74	16	12	...	1	4	25
Fernando Verdasco	Marc Gicquel	1	1	3	0	64	37	36	11	...	1	3	5
Federico Delbonis	Julian Reister	1	1	3	1	55	52	45	27	...	1	8	9
Mikhail Youzhny	Pablo Andujar	1	1	3	1	65	70	35	20	...	3	12	11
Carlos Berlocq	John Isner	1	0	0	3	73	45	27	10	...	4	9	16

Player1	Player2	Round	Result	FNL.1	FNL.2	FSP.1	FSW.1	SSP.1	SSW.1	...	BPC.2	BPW.2	NPA.2
Andrey Kuznetsov	Ryan Harrison	1	0	0	3	71	48	29	11	...	4	8	12
Jack Sock	Guillermo Garcia-Lopez	1	1	3	0	56	32	44	23	...	2	7	9
Tommy Haas	Guillaume Rufin	1	1	3	0	65	48	35	13	...	1	4	14
Jiri Vesely	Philipp Kohlschreiber	1	0	1	3	61	59	39	25	...	3	13	16
Simone Bolelli	Yen-Hsun Lu	1	0	0	3	63	33	37	12	...	4	11	4
Bernard Tomic	Victor Hanescu	1	0	0	3	63	38	37	17	...	3	5	14
Alexandr Dolgopolov	Dmitry Tursunov	1	0	0	3	55	51	45	22	...	4	8	7
Alejandro [illegible]	Grigor [illegible]	1	0	0	2	54	10	46	7	...	2	2	6

Notice how in the above code, we use `desc()` to sort from largest to smallest.

Unforced errors are bad, so we might be interested in finding matches with the fewest unforced errors.
Again we'll use the `select` function because we are only interested in some of the columns.

In [12]:
```
tennis_data %>%
    arrange(UFE.1+UFE.2) %>%
    select(Player1:Result, UFE.1,UFE.2) %>% head() # use head so the whole table d
oesn't print out
```

Player1	Player2	Round	Result	UFE.1	UFE.2
Dmitry Tursunov	Victor Hanescu	2	0	15	10
Alejandro Falla	Grigor Dimitrov	1	0	15	16
Jurgen Zopp	Tommy Robredo	1	0	27	7
Blaz Kavcic	James Duckworth	1	1	8	27
Igor Sijsling	Jurgen Melzer	1	1	16	22
Nick Kyrgios	Marin Cilic	2	0	23	15

Remember that `select` has some helper functions. How could we rewrite the above code using `starts_with` ?

In [13]:
```
tennis_data %>%
    arrange(UFE.1+UFE.2) %>%
    select(Player1:Result, starts_with('UFE')) %>%
    head()
```

Player1	Player2	Round	Result	UFE.1	UFE.2
Dmitry Tursunov	Victor Hanescu	2	0	15	10
Alejandro Falla	Grigor Dimitrov	1	0	15	16
Jurgen Zopp	Tommy Robredo	1	0	27	7
Blaz Kavcic	James Duckworth	1	1	8	27
Igor Sijsling	Jurgen Melzer	1	1	16	22
Nick Kyrgios	Marin Cilic	2	0	23	15

We can also use `contains()`

In [14]:
```
tennis_data %>%
    arrange(UFE.1+UFE.2) %>%
    select(Player1:Result, contains('UFE')) %>%
    head()
```

Player1	Player2	Round	Result	UFE.1	UFE.2
Dmitry Tursunov	Victor Hanescu	2	0	15	10
Alejandro Falla	Grigor Dimitrov	1	0	15	16
Jurgen Zopp	Tommy Robredo	1	0	27	7
Blaz Kavcic	James Duckworth	1	1	8	27
Igor Sijsling	Jurgen Melzer	1	1	16	22
Nick Kyrgios	Marin Cilic	2	0	23	15

Notice that variables corresponding to `Player1` end in `1`. How would we select all the player 1 variables?

In [15]:
```
tennis_data %>%
    select(ends_with('1')) %>%
    head()
```

Player1	FNL.1	FSP.1	FSW.1	SSP.1	SSW.1	ACE.1	DBF.1	WNR.1	UFE.1	BPC.1	BPW.1	NPA.1	NPW.1
Pablo Carreno-Busta	0	62	27	38	11	1	3	12	29	1	3	9	20
Somdev Devvarman	3	62	54	38	22	7	3	26	20	5	8	12	21
Tobias Kamke	3	62	53	38	15	4	6	42	55	10	22	14	32
Julien Benneteau	3	72	87	28	19	14	2	48	27	4	13	14	30
Lukas Lacko	0	52	31	48	22	4	4	21	24	1	1	3	5
Jan Hajek	3	70	58	30	18	4	4	35	36	6	12	8	10

`mutate`

We are likely interested in some aggregate statistics, i.e., combining the results of players 1 and 2 in a match. We'll use `mutate` to create new variables to analyze these statistics.

Suppose we're interested in looking at the length of matches (how many sets are played). One way to do this is to add `FNL1` (total number of sets won by player 1) to `FNL2` (total for player 2).

In [16]:
```
tennis_data_2 = tennis_data %>%
                    mutate(total_sets = FNL.1 + FNL.2)
```

Now we can sort the matches from longest to shortest.

In [17]:
```
tennis_data_2 %>%
    arrange(desc(total_sets)) %>%
    head()
```

Player1	Player2	Round	Result	FNL.1	FNL.2	FSP.1	FSW.1	SSP.1	SSW.1	...	BPW.2	NPA.2	NPW.2	Tl
Tobias Kamke	Paolo Lorenzi	1	1	3	2	62	53	38	15	...	18	19	27	
Adrian Mannarino	Pablo Cuevas	1	0	2	3	63	71	37	38	...	20	14	22	
Gilles Simon	Lleyton Hewitt	1	1	3	2	59	42	41	25	...	10	19	35	
Juan Monaco	Daniel Gimeno-Traver	1	0	2	3	78	85	22	22	...	10	11	18	
Jarkko Nieminen	Paul-Henri Mathieu	1	1	3	2	69	84	31	29	...	15	24	33	
Steve Johnson	Albert Montanes	1	0	2	3	55	53	45	32	...	19	11	16	

Exercises

1. A better measure of match length might be to measure the total number of points played. Compute `total_points` from the variables `TPW.1` and `TPW.2`. Add this to `tennis_data_2`.
2. Create a variable `ace_rate` which is the total number of aces in a match divided by the total number of points played. Add this to `tennis_data_2`.
3. Create a variable `cilic` that is `TRUE` for all matches in which Marin Cilic played and `FALSE` otherwise.
4. Sort the data by `Round`, then by `ace_rate`
5. Create a table containing all matches before the 6th round in which both players had a first serve percentage above 65%
6. A player wins in straight sets if his opponent does not win a single set. How many matches were *not* won in straight sets.

In [18]:
```
#1
tennis_data_2 = tennis_data_2 %>% mutate(total_points = TPW.1 + TPW.2)
```

In [19]:
```
#2
tennis_data_2 = tennis_data_2 %>% mutate(ace_rate = (ACE.1+ACE.2)/total_points)
```

In [20]:
```
#3
mutate(tennis_data_2, cilic=(Player1=='Marin Cilic' | Player2=='Marin Cilic')) %>%
head()
```

Player1	Player2	Round	Result	FNL.1	FNL.2	FSP.1	FSW.1	SSP.1	SSW.1	...	TPW.2	ST1.2	ST2.2	ST3
Pablo Carreno-Busta	Roger Federer	1	0	0	3	62	27	38	11	...	88	6	6	
Somdev Devvarman	Daniel Munoz-De La Nava	1	1	3	0	62	54	38	22	...	106	3	3	
Tobias Kamke	Paolo Lorenzi	1	1	3	2	62	53	38	15	...	139	3	3	
Julien Benneteau	Ricardas Berankis	1	1	3	1	72	87	28	19	...	149	6	3	
Lukas Lacko	Sam Querrey	1	0	0	3	52	31	48	22	...	93	6	6	
Jan Hajek	Denis Kudla	1	1	3	1	70	58	30	18	...	93	2	7	

In [21]:
```
#4
tennis_data_2 %>%
    arrange(Round, ace_rate) %>%
    head()
```

Player1	Player2	Round	Result	FNL.1	FNL.2	FSP.1	FSW.1	SSP.1	SSW.1	...	NPW.2	TPW.2	ST1.2
Jesse Levine	Kei Nishikori	1	0	0	3	52	16	48	9	...	7	86	6
Marinko Matosevic	David Ferrer	1	0	0	3	59	29	41	10	...	11	98	6
Florent Serra	Nikolay Davydenko	1	0	0	3	64	35	36	13	...	5	96	6
Jack Sock	Guillermo Garcia-Lopez	1	1	3	0	56	32	44	23	...	14	66	2
Sergiy Stakhovsky	Richard Gasquet	1	0	0	3	55	22	45	17	...	27	88	6
Blaz Kavcic	James Duckworth	1	1	3	0	63	33	37	17	...	22	50	2

In [22]:
```
#5
tennis_data_2 %>% filter(Round < 6, FSP.1>65, FSP.2>65)
```

Player1	Player2	Round	Result	FNL.1	FNL.2	FSP.1	FSW.1	SSP.1	SSW.1	...	NPW.2	TPW.2	ST1.2	ST
Jan Hajek	Denis Kudla	1	1	3	1	70	58	30	18	...	9	93	2	
Benjamin Becker	Jeremy Chardy	1	0	0	3	66	48	34	13	...	16	108	6	
Tomas Berdych	Gael Monfils	1	0	2	3	66	91	34	33	...	21	189	7	
Michal Przysiezny	Rhyne Williams	1	1	3	1	67	75	33	27	...	16	134	3	
Stanislas Wawrinka	Thiemo De Bakker	1	1	3	1	69	74	31	24	...	33	135	5	
Rafael Nadal	Daniel Brands	1	1	3	1	84	74	16	12	...	40	115	6	
Evgeny Donskoy	Kevin Anderson	2	0	1	3	67	62	33	21	...	19	148	6	
Gael Monfils	Tommy Robredo	3	0	2	3	68	73	32	27	...	40	170	2	
Nikolay Davydenko	Richard Gasquet	3	0	0	3	76	46	24	9	...	31	101	6	
Rafael Nadal	Fabio Fognini	3	1	3	0	78	52	22	14	...	31	111	6	
Novak Djokovic	Grigor Dimitrov	3	1	3	0	68	39	32	13	...	12	61	2	
Jo-Wilfried Tsonga	Roger Federer	5	1	3	0	75	46	25	10	...	30	73	5	

In [23]:
```
#6
tennis_data_2 %>% filter(FNL.1 > 0, FNL.2 > 0) %>% nrow()
```

55

summarise

You'll learn more about data summaries in this week's lecture, but we'll introduce the concept here.

In [24]:
```
tennis_data_2 %>%
    summarise(total_matches=n(), avg_points = mean(total_points), avg_sets = mean
(total_sets))
```

total_matches	avg_points	avg_sets
125	219.44	3.568

We can combine the summarise operation with other operations from `dplyr`

In [25]:
```
tennis_data_2 %>%
    group_by(Round) %>%
    summarise(total_matches=n(), avg_points = mean(total_points))
```

```
`summarise()` ungrouping output (override with `.groups` argument)
```

Round	total_matches	avg_points
1	63	214.5714
2	31	228.5484
3	16	225.5000
4	8	231.8750
5	4	166.5000
6	2	263.0000
7	1	172.0000

In [26]:
```
usa_players = c('Sam Querrey', 'John Isner')
tennis_data_2 %>%
    group_by(Player1 %in% usa_players | Player2 %in% usa_players) %>%
    summarise(avg_ace = mean(ace_rate))
```

```
`summarise()` ungrouping output (override with `.groups` argument)
```

Player1 %in% usa_players | Player2 %in% usa_players	avg_ace
FALSE	0.05600242
TRUE	0.07445462

We can even sort the summary table based on the results of the summary statistics

In [27]:
```
tennis_data_2 %>%
    filter(Round < 5) %>%
    group_by(Round) %>%
    summarise(avg_FSP = mean((FSP.1 + FSP.2)/2)) %>%
    arrange(desc(avg_FSP))
```

```
`summarise()` ungrouping output (override with `.groups` argument)
```

Round	avg_FSP
3	64.71875
4	63.93750
2	62.59677
1	61.42857

We can assign summary tables to variables and then plot them.

In [28]:
```
round = tennis_data_2 %>%
        filter(total_sets > 2) %>%
        group_by(Round) %>%
        summarise(avg_ace = mean(ace_rate), avg_points = mean(total_points))
```

```
`summarise()` ungrouping output (override with `.groups` argument)
```

In [29]:
```
ggplot(round) +
    geom_bar(aes(x=Round, y=avg_points),stat='identity', fill='green')
```