Activity 4

Figure 3.16:

```
library(NHANES)
ggplot(
  data = slice_sample(NHANES, n = 1000),
  aes(x = Age, y = Height, color = fct_relevel(Gender, "male"))
) +
  geom_point() +
  geom_smooth() +
  xlab("Age (years)") +
  ylab("Height (cm)") +
  labs(color = "Gender")
```

Figure 3.17

```
library(macleish)
ggplot(data = whately_2015, aes(x = when, y = temperature)) +
  geom_line(color = "darkgray") +
  geom_smooth() +
  xlab(NULL) +
  ylab("Temperature (degrees Celsius)")
```

Figure 3.18:

```
whately_2015 %>%
  mutate(month = as.factor(lubridate::month(when, label = TRUE))) %>%
  group_by(month) %>%
  skim(temperature) %>%
  select(-na)
```

```
ggplot(
  data = whately_2015,
  aes(
    x = lubridate::month(when, label = TRUE),
    y = temperature
  )
) +
  geom_boxplot() +
  xlab("Month") +
  ylab("Temperature (degrees Celsius)")
```

Figure 3.19:

```
library(NHANES)
library(ggmosaic)
library(tidyverse)
library(ggplot2)

ggplot(data = NHANES) +
  geom_mosaic(aes(x = product(AgeDecade, BMI), fill = Diabetes)) +
  labs(x = "BMI", y = "Age (by decade)")
```

Activity 5

```
library(Lahman)
team <- Teams
astros <- Teams %>%
  select(yearID, teamID, W, L, R, RA) %>%
  filter(teamID == "HOU" & yearID %in% 2004:2012) %>%
```

rename(RS = R) %>%

mutate(WPct = W / (W + L)) %>%

mutate(WPct_hat = 1 / (1 + (RA/RS)^2))%>%

mutate(W_hat = WPct_hat * (W + L))

astros

	yearID	teamID	W	L	RS	RA	WPct	WPct_hat	W_hat
1	2004	HOU	92	70	803	698	0.5679012	0.5696127	92.27726
2	2005	HOU	89	73	693	609	0.5493827	0.5642487	91.40829
3	2006	HOU	82	80	735	719	0.5061728	0.5110028	82.78245
4	2007	HOU	73	89	723	813	0.4506173	0.4416067	71.54029
5	2008	HOU	86	75	712	743	0.5341615	0.4787038	77.07132
6	2009	HOU	74	88	643	770	0.4567901	0.4108406	66.55617
7	2010	HOU	76	86	611	729	0.4691358	0.4126179	66.84410
8	2011	HOU	56	106	615	796	0.3456790	0.3737988	60.55541
9	2012	HOU	55	107	583	794	0.3395062	0.3502837	56.74595

I then piped this to find out which seasons are the best for the Astros: arrange(astros, desc(WPct))

	yearID	teamID	W	L	RS	RA	WPct	WPct_hat	W_hat
1	2004	HOU	92	70	803	698	0.5679012	0.5696127	92.27726
2	2005	HOU	89	73	693	609	0.5493827	0.5642487	91.40829
3	2008	HOU	86	75	712	743	0.5341615	0.4787038	77.07132
4	2006	HOU	82	80	735	719	0.5061728	0.5110028	82.78245
5	2010	HOU	76	86	611	729	0.4691358	0.4126179	66.84410
6	2009	HOU	74	88	643	770	0.4567901	0.4108406	66.55617
7	2007	HOU	73	89	723	813	0.4506173	0.4416067	71.54029
8	2011	HOU	56	106	615	796	0.3456790	0.3737988	60.55541
9	2012	HOU	55	107	583	794	0.3395062	0.3502837	56.74595

I then piped this to find the Astros' most lucky season:

mutate(Diff = W - W_hat) %>%

arrange(desc(Diff))

```
  yearID teamID  W   L  RS  RA      WPct  WPct_hat    W_hat       Diff
1   2010    HOU 76  86 611 729 0.4691358 0.4126179 66.84410  9.1558996
2   2008    HOU 86  75 712 743 0.5341615 0.4787038 77.07132  8.9286841
3   2009    HOU 74  88 643 770 0.4567901 0.4108406 66.55617  7.4438271
4   2007    HOU 73  89 723 813 0.4506173 0.4416067 71.54029  1.4597102
5   2004    HOU 92  70 803 698 0.5679012 0.5696127 92.27726 -0.2772601
6   2006    HOU 82  80 735 719 0.5061728 0.5110028 82.78245 -0.7824527
7   2012    HOU 55 107 583 794 0.3395062 0.3502837 56.74595 -1.7459542
8   2005    HOU 89  73 693 609 0.5493827 0.5642487 91.40829 -2.4082902
9   2011    HOU 56 106 615 796 0.3456790 0.3737988 60.55541 -4.5554134
```

It seems that they were the luckiest in 2010, 2008, 2009, and 2007.

Afterwards, I piped this to find the statistics about their performance: skim(W)

```
── Variable type: numeric ──────────────────
  var n na mean   sd p0 p25 p50 p75 p100
1 W   9  0 75.9 13.3 55  73  76  86   92
```

It seems that the Astros wins, on average, about 76 games per season.

To find all of the statistics, I piped this:

```
summarize(
   num_years = n(),
   total_W = sum(W),
   total_L = sum(L),
   total_WPct = sum(W) / sum(W + L),
   sum_resid = sum(W - W_hat)
 )
```

```
  num_years total_W total_L total_WPct sum_resid
1         9     683     774  0.4687714  17.21875
```

Activity 6

```
library(Lahman)
ruth <- Batting %>%
  filter(playerID == "ruthba01")%>%
  summarize(
    span = paste(min(yearID), max(yearID), sep = "-"),
```

```
    num_years = n_distinct(yearID),
    num_teams = n_distinct(teamID),
    BA = sum(H)/sum(AB),
    tH = sum(H),
    tHR = sum(HR),
    tRBI = sum(RBI)
  )
ruth
```

```
        span num_years num_teams        BA   tH tHR tRBI
1 1914-1935        22          3 0.3421053 2873 714 2217
```

```
library(Lahman)
ruth <- Batting %>%
  filter(playerID == "ruthba01") %>%
  group_by(teamID) %>%
  summarize(
    span = paste(min(yearID), max(yearID), sep = "-"),
    num_years = n_distinct(yearID),
    num_teams = n_distinct(teamID),
    BA = sum(H)/sum(AB),
    tH = sum(H),
    tHR = sum(HR),
    tRBI = sum(RBI)
  ) %>%
  arrange(span)
ruth
```

```
# A tibble: 3 × 8
  teamID span      num_years num_teams    BA    tH   tHR  tRBI
  <fct>  <chr>         <int>     <int> <dbl> <int> <int> <int>
1 BOS    1914-1919         6         1 0.308   342    49   230
2 NYA    1920-1934        15         1 0.349  2518   659  1975
3 BSN    1935-1935         1         1 0.181    13     6    12
```

```
library(Lahman)
ruth <- Batting %>%
  filter(playerID == "ruthba01") %>%
  group_by(lgID) %>%
  summarize(
    span = paste(min(yearID), max(yearID), sep = "-"),
    num_years = n_distinct(yearID),
    num_teams = n_distinct(teamID),
    BA = sum(H)/sum(AB),
    tH = sum(H),
    tHR = sum(HR),
    tRBI = sum(RBI)
  ) %>%
  arrange(span)
ruth
```

```
# A tibble: 2 × 8
  lgID  span      num_years num_teams    BA    tH   tHR  tRBI
  <fct> <chr>         <int>     <int> <dbl> <int> <int> <int>
1 AL    1914-1934        21         2 0.344  2860   708  2205
2 NL    1935-1935         1         1 0.181    13     6    12
```

```
library(Lahman)
ruth <- Batting %>%
  filter(playerID == "ruthba01") %>%
  group_by(yearID) %>%
  summarize(tHR = sum(HR)) %>%
  filter(tHR >= 30) %>%
```

```
  nrow()
```

ruth

```
[1] 13
```

```
library(Lahman)

People %>%
  filter(nameLast == "Ruth" & nameFirst == "Babe")
```

```
  playerID birthYear birthMonth birthDay birthCountry birthState birthCity deathYear
1 ruthba01      1895          2        6          USA         MD Baltimore      1948
  deathMonth deathDay deathCountry deathState deathCity nameFirst nameLast
1          8       16          USA         NY  New York      Babe     Ruth
      nameGiven weight height bats throws      debut  finalGame  retroID  bbrefID
1 George Herman    215     74    L      L 1914-07-11 1935-05-30 ruthb101 ruthba01
   deathDate  birthDate
1 1948-08-16 1895-02-06
```

```
library(Lahman)

Batting %>%
  filter(playerID == "ruthba01") %>%
  inner_join(People, by = c("playerID" = "playerID")) %>%
  group_by(yearID) %>%
  summarize(
    Age = max(yearID - birthYear),
    num_teams = n_distinct(teamID),
    BA = sum(H)/sum(AB),
    tH = sum(H),
    tHR = sum(HR),
    tRBI = sum(RBI)
  ) %>%
  arrange(yearID)
```

	yearID	Age	num_teams	BA	tH	tHR	tRBI
	<int>	<int>	<int>	<dbl>	<int>	<int>	<int>
1	1914	19	1	0.2	2	0	2
2	1915	20	1	0.315	29	4	21
3	1916	21	1	0.272	37	3	15
4	1917	22	1	0.325	40	2	12
5	1918	23	1	0.300	95	11	66
6	1919	24	1	0.322	139	29	114
7	1920	25	1	0.376	172	54	137
8	1921	26	1	0.378	204	59	171
9	1922	27	1	0.315	128	35	99
10	1923	28	1	0.393	205	41	131

```
library(Lahman)
Batting %>%
  filter(playerID == "ruthba01") %>%
  inner_join(People, by = c("playerID" = "playerID"))  %>%
  group_by(yearID) %>%
  summarize(
    Age = max(yearID - birthYear),
    num_teams = n_distinct(teamID),
    BA = sum(H)/sum(AB),
    tH = sum(H),
    tHR = sum(HR),
    tRBI = sum(RBI),
    OBP = sum(H + BB + HBP) / sum(AB + BB + SF + HBP),
    SLG = sum(H + X2B + 2 * X3B + 3 * HR) / sum(AB)
  ) %>%
  mutate(OPS = OBP + SLG) %>%
  arrange(desc(OPS))
```

	yearID	Age	num_teams	BA	tH	tHR	tRBI	OBP	SLG	OPS
	<int>	<int>	<int>	<dbl>	<int>	<int>	<int>	<dbl>	<dbl>	<dbl>
1	1914	19	1	0.2	2	0	2	NA	0.3	NA
2	1915	20	1	0.315	29	4	21	NA	0.576	NA
3	1916	21	1	0.272	37	3	15	NA	0.419	NA
4	1917	22	1	0.325	40	2	12	NA	0.472	NA
5	1918	23	1	0.300	95	11	66	NA	0.555	NA
6	1919	24	1	0.322	139	29	114	NA	0.657	NA
7	1920	25	1	0.376	172	54	137	NA	0.849	NA
8	1921	26	1	0.378	204	59	171	NA	0.846	NA
9	1922	27	1	0.315	128	35	99	NA	0.672	NA
0	1923	28	1	0.393	205	41	131	NA	0.764	NA

```
library(Lahman)
ruth_by_season <- Batting %>%
  filter(playerID == "ruthba01") %>%
  inner_join(People, by = c("playerID" = "playerID"))  %>%
  group_by(yearID) %>%
  summarize(
    Age = max(yearID - birthYear),
    num_teams = n_distinct(teamID),
    BA = sum(H)/sum(AB),
    tH = sum(H),
    tHR = sum(HR),
    tRBI = sum(RBI),
    OBP = sum(H + BB + HBP) / sum(AB + BB + SF + HBP),
    SLG = sum(H + X2B + 2 * X3B + 3 * HR) / sum(AB)
  ) %>%
  mutate(OPS = OBP + SLG) %>%
  arrange(desc(OPS))

mlb <- Batting %>%
  filter(yearID %in% 1914:1935)%>%
  group_by(yearID) %>%
```

```
  summarize(
    lg_OBP = sum(H + BB + HBP, na.rm = TRUE) /
      sum(AB + BB + SF + HBP, na.rm = TRUE),
    lg_SLG = sum(H + X2B + 2*X3B + 3*HR, na.rm = TRUE) /
      sum(AB, na.rm = TRUE)
  ) %>%
  mutate(lg_OPS = lg_OBP + lg_SLG)


ruth_ratio <- ruth_by_season %>%
  inner_join(mlb, by = c("yearID" = "yearID")) %>%
  mutate(OPS_plus = OPS / lg_OPS) %>%
  select(yearID, Age, OPS, lg_OPS, OPS_plus) %>%
  arrange(desc(OPS_plus)) %>%
```

ruth_ratio

	yearID	Age	OPS	lg_OPS	OPS_plus
	<int>	<int>	<dbl>	<dbl>	<dbl>
1	1914	19	NA	Inf	NA
2	1915	20	NA	Inf	NA
3	1916	21	NA	Inf	NA
4	1917	22	NA	Inf	NA
5	1918	23	NA	Inf	NA
6	1919	24	NA	Inf	NA
7	1920	25	NA	Inf	NA
8	1921	26	NA	Inf	NA
9	1922	27	NA	Inf	NA
10	1923	28	NA	Inf	NA

Note: It seems that OPS data has not been recorded when Ruth was playing, which is from 1914 to 1935

Activity 7

nest() – collapse all ungrouped variables in a data frame into a tibble

pull() – extract a single column

purr::pluck() – access any of tables inside of the list of all tables. Typically used to extract tables from website/html.

unnest() – undo the nesting structure of a column

Activity 8

```
library(lubridate)
library(tidyverse)
library(rvest)
tables <- "http://en.wikipedia.org/wiki/List_of_nuclear_reactors" %>%
  read_html() %>%
  html_nodes(css = "table")

idx <- tables %>%
  html_text() %>%
  str_detect("Fukushima Daiichi") %>%
  which()

reactors <- tables %>%
  purrr::pluck(idx) %>%
  html_table(fill = TRUE) %>%
  janitor::clean_names() %>%
  rename(
    reactor_type = reactor,
    reactor_model = reactor_2,
    capacity_net = net_capacity_mw,
  ) %>%
```

```
  tail(-1)


glimpse(reactors)
```

```
Columns: 9
$ name                <chr> "Fugen", "Fukushima Daiichi", "Fukushima Daiichi", "Fuk…
$ unit_no             <chr> "1", "1", "2", "3", "4", "5", "6", "1", "2", "3", "4", …
$ reactor_type        <chr> "HWLWR", "BWR", "BWR", "BWR", "BWR", "BWR", "BWR", "BWR…
$ reactor_model       <chr> "ATR", "BWR-3", "BWR-4", "BWR-4", "BWR-4", "BWR-4", "BW…
$ status              <chr> "Shut down", "Inoperable", "Inoperable", "Inoperable", …
$ capacity_net        <chr> "148", "439", "760", "760", "760", "760", "1067", "1067…
$ construction_start  <chr> "10 May 1972", "25 July 1967", "9 June 1969", "28 Decem…
$ commercial_operation <chr> "20 March 1979", "26 March 1971", "18 July 1974", "27 M…
$ closure             <chr> "29 March 2003", "19 May 2011", "19 May 2011", "19 May …
```

```
reactors <- reactors %>%
  mutate(
    plant_status = ifelse(
      str_detect(status, "Shut down"),
      "Shut down", "Not formally shut down"
    ),
    capacity_net = parse_number(capacity_net),
    construct_date = dmy(construction_start),
    operation_date = dmy(commercial_operation),
    closure_date = dmy(closure)
  )
glimpse(reactors)
```

```
$ name                <chr> "Fugen", "Fukushima Daiichi", "Fukushima Daiichi", "Fuk…
$ unit_no             <chr> "1", "1", "2", "3", "4", "5", "6", "1", "2", "3", "4", …
$ reactor_type        <chr> "HWLWR", "BWR", "BWR", "BWR", "BWR", "BWR", "BWR", "BWR…
$ reactor_model       <chr> "ATR", "BWR-3", "BWR-4", "BWR-4", "BWR-4", "BWR-4", "BW…
$ status              <chr> "Shut down", "Inoperable", "Inoperable", "Inoperable", …
$ capacity_net        <dbl> 148, 439, 760, 760, 760, 760, 1067, 1067, 1067, 1067, 1…
$ construction_start  <chr> "10 May 1972", "25 July 1967", "9 June 1969", "28 Decem…
$ commercial_operation <chr> "20 March 1979", "26 March 1971", "18 July 1974", "27 M…
$ closure             <chr> "29 March 2003", "19 May 2011", "19 May 2011", "19 May …
$ plant_status        <chr> "Shut down", "Not formally shut down", "Not formally sh…
$ construct_date      <date> 1972-05-10, 1967-07-25, 1969-06-09, 1970-12-28, 1973-0…
$ operation_date      <date> 1979-03-20, 1971-03-26, 1974-07-18, 1976-03-27, 1978-1…
$ closure_date        <date> 2003-03-29, 2011-05-19, 2011-05-19, 2011-05-19, 2011-0…
```

```
ggplot(
  data = reactors,
  aes(x = construct_date, y = capacity_net, color = plant_status
  )
) +
  geom_point() +
  geom_smooth() +
  xlab("Date of Plant Construction") +
  ylab("Net Plant Capacity (MW)")
```

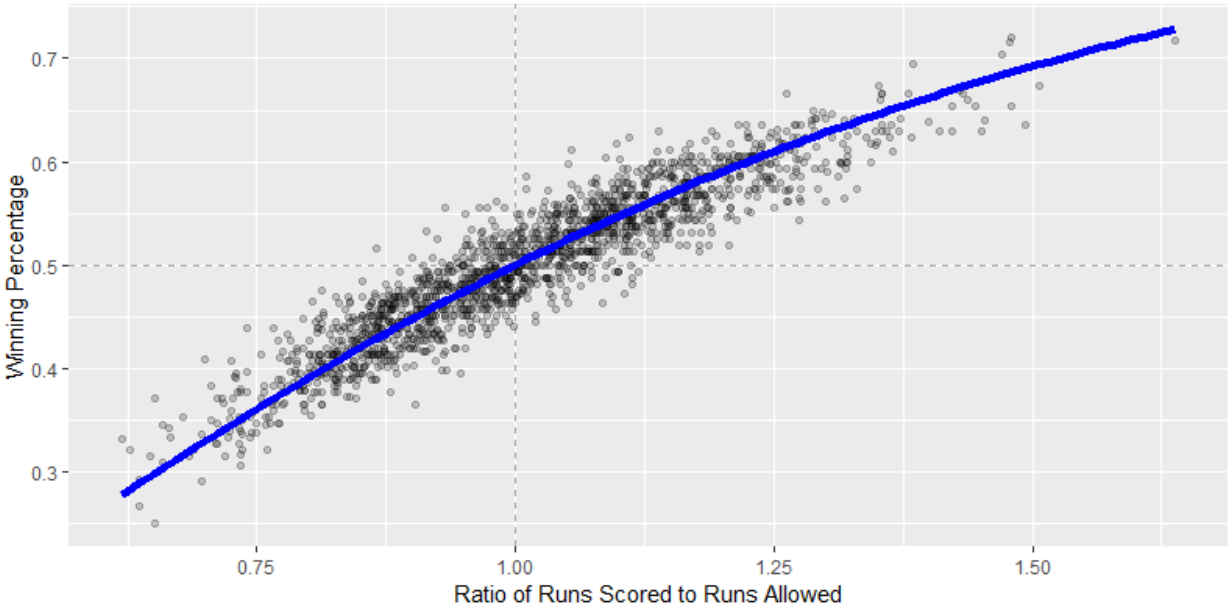
Activity 9

```
exp_wpct <- function(x) {
  return(1/(1 + (1/x)^2))
}

TeamRuns <- Teams %>%
  filter(yearID >= 1954) %>%
  rename(RS = R) %>%
```

```
  mutate(WPct = W / (W + L), run_ratio = RS/RA) %>%
  select(yearID, teamID, lgID, WPct, run_ratio)

ggplot(data = TeamRuns, aes(x = run_ratio, y = WPct)) +
  geom_vline(xintercept = 1, color = "darkgray", linetype = 2) +
  geom_hline(yintercept = 0.5, color = "darkgray", linetype = 2) +
  geom_point(alpha = 0.2) +
  stat_function(fun = exp_wpct, size = 2, color = "blue") +
  xlab("Ratio of Runs Scored to Runs Allowed") +
  ylab("Winning Percentage")
```




```
TeamRuns %>%
  nls(
    formula = WPct ~ 1/(1 + (1/run_ratio)^k),
    start = list(k = 2)
  ) %>%
  coef()
```

```
        k
1.835093
```

The nls() function finds the nonlinear least-squares estimates of a nonlinear model for the entered parameter.

```
fit_k <- function(x) {
  mod <- nls(
    formula = WPct ~ 1/(1 + (1/run_ratio)^k),
    data = x,
    start = list(k = 2)
  )
  return(tibble(k = coef(mod), n = nrow(x)))
}
```

```
fit_k(TeamRuns)
```

```
# A tibble: 1 × 2
      k     n
  <dbl> <int>
1  1.84  1738
```

```
TeamRuns %>%
  mutate(decade = yearID %/% 10 * 10) %>%
  group_by(decade) %>%
  group_modify(~fit_k(.x))
```

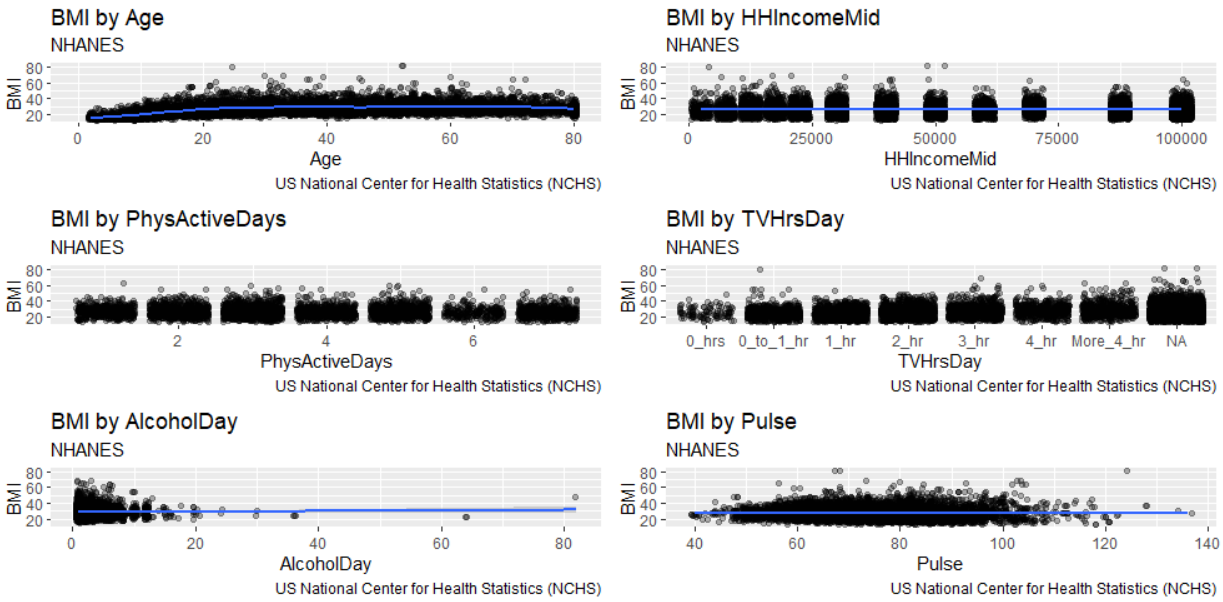
	decade	k	n
	<dbl>	<dbl>	<int>
1	1950	1.69	96
2	1960	1.90	198
3	1970	1.74	246
4	1980	1.93	260
5	1990	1.88	278
6	2000	1.94	300
7	2010	1.77	300
8	2020	1.81	60

```
library(NHANES)
ggplot(NHANES, aes(x = Age, y = BMI)) +
  geom_point() +
  geom_smooth()

bmi_plot <- function(.data, x_var) {
  ggplot(.data, aes(y = BMI)) +
    aes_string(x = x_var) +
    geom_jitter(alpha = 0.3) +
    geom_smooth() +
    labs(
      title = paste("BMI by", x_var),
      subtitle = "NHANES",
      caption = "US National Center for Health Statistics (NCHS)"
    )
}

c("Age", "HHIncomeMid", "PhysActiveDays",
  "TVHrsDay", "AlcoholDay", "Pulse") %>%
  map(bmi_plot, .data = NHANES) %>%
```

patchwork::wrap_plots(ncol = 2)



Activity 10

- There are concerns that all these rules and regulations can threaten their job by making them look incompetent if the data scientist appears to not fully understand. This is an especially important reason since data analysis is such a competitive field – makes number 1 from the data science oath difficult.
- There's no direct incentive to take the oath.
- People might not want to publish their metadata to have a competitive advantage over other data scientists – number 3 is difficult to follow.
- Data collection and organization is a painstaking process – number 4 is difficult to follow.
- Number 11 is difficult to follow in some organization as it would take a lot of work to ensure the privacy and security on our owns, especially since senior staffs are not as likely to care about. There are usually teams at some organizations simply for this purpose for a reason.