

Lab 1

Rhea Abraham, worked with Varsha Murali

Due on 02/07 at 11:59 PM

Instructions: This lab report needs to be professional. Only report relevant and finalized code. Your writing should be concise and void of spelling errors. Use code chunk options to hide unnecessary messages/warnings. Your report should be reproducible. Reports that involve simulations need to have the random seed specified so that simulation results are reproducible. You are allowed to work on this lab assignment in groups of 2-3. You still need to submit an individual lab report if you do work in a group, and you need to list your collaborators.

Question 1 In lecture it was demonstrated that baseball is a game of offense, pitching, and defense with a regression model that considered expected run differential as a function of explanatory variables OPS, WHIP, and FP. Do the following:

- Fit a similar regression model with runs as the response variable. Report problems with this model. Investigate problematic residuals to discover what went wrong. Fix the problem with this model by adding categorical variable(s) to the list of explanatory variables. Briefly explain what went wrong.
- We can significantly improve the regression model in the notes through a principled rescaling of OPS, WHIP, and FP. Split the Teams data frame by `yearID` and, for each year, create variables `OPSscale = OPS/avgOPS`, `WHIPscale = avgWHIP/WHIP`, and `FPscale = FP/avgFP` which require you to first create league average variables `avgOPS`, `avgWHIP`, and `avgFP`. Fit the linear regression model with runs differential as the response and explanatory variables `OPSscale`, `WHIPscale`, and `FPscale`, and report relevant output. Why does this model perform so much better than the model in the notes? Support your answer.

```
library(Lahman)
data(Batting)
data(Pitching)
data(Fielding)
data(Teams)
```

```
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
```

```
library(ggplot2)
library(retrosheet)
```

```
##
## For Retrosheet data obtained with this package:
##
## The information used here was obtained free of charge from
## and is copyrighted by Retrosheet. Interested parties may
## contact Retrosheet at "www.retrosheet.org"

library(baseballr)

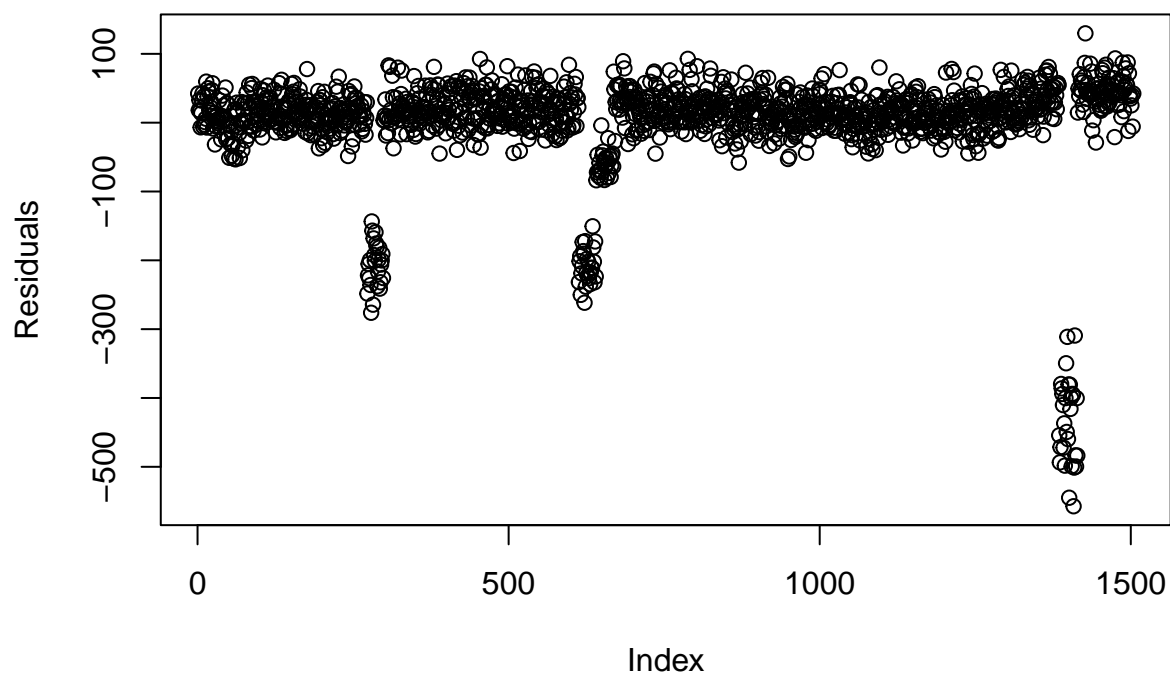
Teams <- Teams %>%
  mutate(
    OBP = (H + BB + HBP) / (AB + BB + HBP + SF),
    SLG = (H + X2B + (2 * X3B) + (3 * HR)) / AB,
    OPS = OBP + SLG,
    WHIP = (BB + H) / (IPouts / 3)
  )

model1 <- lm(R ~ OPS + WHIP + FP, data = Teams)
summary(model1)

##
## Call:
## lm(formula = R ~ OPS + WHIP + FP, data = Teams)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -557.30   -3.74   14.88   34.01  129.78
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -678.966     615.873  -1.102   0.270
## OPS           1469.084     100.961  14.551 < 2e-16 ***
## WHIP           232.833      49.781   4.677 3.17e-06 ***
## FP             -3.085     635.920  -0.005   0.996
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 80.4 on 1500 degrees of freedom
## (1541 observations deleted due to missingness)
## Multiple R-squared:  0.5004, Adjusted R-squared:  0.4994
## F-statistic: 500.9 on 3 and 1500 DF,  p-value: < 2.2e-16

plot(model1$residuals, main = "Residuals of Initial Model", ylab = "Residuals")
```

Residuals of Initial Model



```
Teams_clean <- Teams %>% filter(complete.cases(OPS, WHIP, FP, R))
Teams_clean <- Teams_clean %>% mutate(residuals = model1$residuals) %>% arrange(desc(abs(residuals))) %>%
print(Teams_clean)
```

##	yearID	lgID	teamID	franchID	divID	Rank	G	Ghome	W	L	DivWin	WCWin	LgWin
## 1	2020	NL	NYN	NYM	E	4	60	29	26	34	N	N	N
## 2	2020	NL	ATL	ATL	E	1	60	30	35	25	Y	N	N
## 3	2020	NL	PHI	PHI	E	3	60	32	28	32	N	N	N
## 4	2020	NL	SFN	SFG	W	3	60	33	29	31	N	N	N
## 5	2020	NL	LAN	LAD	W	1	60	30	43	17	Y	N	Y
## 6	2020	AL	NYA	NYN	E	2	60	31	33	27	N	Y	N
## 7	2020	AL	BOS	BOS	E	5	60	31	24	36	N	N	N
## 8	2020	NL	WAS	WSN	E	4	60	33	26	34	N	N	N
## 9	2020	NL	SDN	SDP	W	2	60	32	37	23	N	Y	N
## 10	2020	AL	LAA	ANA	W	4	60	31	26	34	N	N	N

##	WSWin	R	AB	H	X2B	X3B	HR	BB	SO	SB	CS	HBP	SF	RA	ER	ERA	CG	SHO	SV
## 1	N	286	2023	551	106	7	86	197	498	20	10	45	13	308	284	4.98	0	1	11
## 2	N	348	2074	556	130	3	103	239	573	23	4	23	7	288	257	4.41	0	4	13
## 3	N	306	1948	500	90	10	82	229	480	35	8	28	10	311	284	5.14	3	5	11
## 4	N	299	2019	532	107	14	81	195	499	19	8	27	13	297	267	4.64	1	0	13
## 5	Y	349	2042	523	97	6	118	228	471	29	8	30	12	213	181	3.02	0	5	15
## 6	N	315	1915	473	87	7	94	251	480	27	7	30	11	270	242	4.35	2	2	14
## 7	N	292	2083	552	118	7	81	187	545	31	9	21	9	351	325	5.58	0	2	14
## 8	N	293	1968	519	112	12	66	192	451	33	12	32	21	301	285	5.09	2	3	12
## 9	N	325	1972	506	103	12	95	204	479	55	13	28	14	241	223	3.86	1	5	13
## 10	N	294	2020	501	97	8	85	239	490	21	8	25	18	321	297	5.09	1	2	12

##	IPouts	HA	HRA	BBA	SOA	E	DP	FP	name
## 1	1540	511	81	219	574	32	39	0.985	New York Mets
## 2	1573	494	69	220	506	33	52	0.985	Atlanta Braves
## 3	1491	550	80	185	532	35	57	0.983	Philadelphia Phillies

```
## 4 1553 474 69 210 488 41 43 0.980 San Francisco Giants
## 5 1616 424 66 145 517 40 46 0.982 Los Angeles Dodgers
## 6 1502 455 83 168 528 48 37 0.976 New York Yankees
## 7 1572 587 98 252 537 45 59 0.979 Boston Red Sox
## 8 1511 548 94 216 508 39 48 0.981 Washington Nationals
## 9 1561 456 70 170 565 32 46 0.985 San Diego Padres
## 10 1576 492 82 199 523 36 36 0.983 Los Angeles Angels of Anaheim
## park attendance BPF PPF teamIDBR teamIDlahman45
## 1 Citi Field 0 94 94 NYM NYN
## 2 SunTrust Park 0 107 106 ATL ATL
## 3 Citizens Bank Park 0 101 101 PHI PHI
## 4 Oracle Park 0 94 95 SFG SFN
## 5 Dodger Stadium 0 97 94 LAD LAN
## 6 Yankee Stadium III 0 98 96 NYY NYA
## 7 Fenway Park II 0 105 105 BOS BOS
## 8 Nationals Park 0 103 102 WSN MON
## 9 Petco Park 0 94 95 SDP SDN
## 10 Angel Stadium of Anaheim 0 100 101 LAA ANA
## teamIDretro OBP SLG OPS WHIP residuals
## 1 NYN 0.3481124 0.4592190 0.8073314 1.457143 -557.3032
## 2 ATL 0.3491251 0.4826422 0.8317673 1.516211 -544.9547
## 3 PHI 0.3417607 0.4394251 0.7811858 1.466801 -501.1480
## 4 SFN 0.3345164 0.4507182 0.7852346 1.404379 -499.5714
## 5 LAN 0.3378028 0.4828599 0.8206627 1.394183 -499.2382
## 6 NYA 0.3416402 0.4469974 0.7886376 1.446072 -498.2907
## 7 BOS 0.3304348 0.4450312 0.7754660 1.410305 -493.6035
## 8 WAS 0.3357433 0.4334350 0.7691783 1.411648 -483.6728
## 9 SDN 0.3327322 0.4655172 0.7982494 1.364510 -483.3931
## 10 ANA 0.3323197 0.4301980 0.7625177 1.408629 -472.1789
```

```
model2 <- lm(R ~ OPS + WHIP + FP + factor(yearID) + factor(lgID), data = Teams)
summary(model2)
```

```
##
## Call:
## lm(formula = R ~ OPS + WHIP + FP + factor(yearID) + factor(lgID),
## data = Teams)
##
## Residuals:
## Min 1Q Median 3Q Max
## -121.758 -16.179 -0.625 15.774 130.531
##
## Coefficients:
## Estimate Std. Error t value Pr(>|t|)
## (Intercept) -914.7783 246.2386 -3.715 0.000211 ***
## OPS 1583.8870 40.3503 39.253 < 2e-16 ***
## WHIP 177.7173 19.7382 9.004 < 2e-16 ***
## FP 259.7588 254.3895 1.021 0.307375
## factor(yearID)1971 -18.7843 7.4105 -2.535 0.011355 *
## factor(yearID)1972 -40.9070 7.4417 -5.497 4.56e-08 ***
## factor(yearID)1973 -8.0063 7.3751 -1.086 0.277848
## factor(yearID)1974 -3.8910 7.3957 -0.526 0.598886
## factor(yearID)1975 -9.1275 7.4077 -1.232 0.218085
## factor(yearID)1976 -2.6751 7.4009 -0.361 0.717815
## factor(yearID)1977 -13.1326 7.2481 -1.812 0.070215 .
```

```

## factor(yearID)1978 -21.8414      7.2314 -3.020 0.002569 **
## factor(yearID)1979 -12.4762      7.2379 -1.724 0.084971 .
## factor(yearID)1980 -13.6271      7.2294 -1.885 0.059636 .
## factor(yearID)1981 -232.4101     7.2538 -32.040 < 2e-16 ***
## factor(yearID)1982 -7.5106       7.2380 -1.038 0.299598
## factor(yearID)1983 -8.5068       7.2337 -1.176 0.239790
## factor(yearID)1984 -6.5044       7.2304 -0.900 0.368485
## factor(yearID)1985 -5.8731       7.2411 -0.811 0.417455
## factor(yearID)1986 -5.9130       7.2401 -0.817 0.414236
## factor(yearID)1987 -1.5062       7.2867 -0.207 0.836271
## factor(yearID)1988 -2.1722       7.2731 -0.299 0.765240
## factor(yearID)1989 -2.5091       7.2506 -0.346 0.729355
## factor(yearID)1990 -12.0333      7.2541 -1.659 0.097370 .
## factor(yearID)1991  1.3341       7.2699  0.184 0.854423
## factor(yearID)1992 -13.9540      7.2854 -1.915 0.055648 .
## factor(yearID)1993 -2.7919       7.1369 -0.391 0.695717
## factor(yearID)1994 -236.6294     7.2058 -32.839 < 2e-16 ***
## factor(yearID)1995 -86.8077     7.1879 -12.077 < 2e-16 ***
## factor(yearID)1996  8.2991       7.2113  1.151 0.249988
## factor(yearID)1997 -13.3152     7.2032 -1.849 0.064731 .
## factor(yearID)1998 -5.4517       7.1148 -0.766 0.443650
## factor(yearID)1999 -6.0313       7.1447 -0.844 0.398719
## factor(yearID)2000 -3.8355       7.1634 -0.535 0.592439
## factor(yearID)2001 -10.0250      7.2061 -1.391 0.164386
## factor(yearID)2002 -18.7459      7.1641 -2.617 0.008972 **
## factor(yearID)2003 -11.5873      7.1959 -1.610 0.107559
## factor(yearID)2004 -14.6691      7.2145 -2.033 0.042205 *
## factor(yearID)2005 -22.9757      7.2257 -3.180 0.001505 **
## factor(yearID)2006 -17.8600      7.2446 -2.465 0.013806 *
## factor(yearID)2007 -10.7948      7.2018 -1.499 0.134119
## factor(yearID)2008 -18.3743      7.1929 -2.555 0.010735 *
## factor(yearID)2009 -25.9287      7.2243 -3.589 0.000343 ***
## factor(yearID)2010 -19.4898      7.1708 -2.718 0.006648 **
## factor(yearID)2011 -16.8349      7.2150 -2.333 0.019768 *
## factor(yearID)2012 -16.4533      7.2694 -2.263 0.023761 *
## factor(yearID)2013 -24.3991      7.3261 -3.330 0.000889 ***
## factor(yearID)2014 -14.0973      7.2880 -1.934 0.053270 .
## factor(yearID)2015 -22.1163      7.3689 -3.001 0.002734 **
## factor(yearID)2016 -18.5805      7.3568 -2.526 0.011655 *
## factor(yearID)2017 -11.8427      7.3776 -1.605 0.108665
## factor(yearID)2018 -0.6412       7.3638 -0.087 0.930629
## factor(yearID)2019  8.1382       7.4656  1.090 0.275853
## factor(yearID)2020 -465.1263     7.3100 -63.629 < 2e-16 ***
## factor(yearID)2021  12.6941      7.4229  1.710 0.087455 .
## factor(yearID)2022  12.4860      7.4333  1.680 0.093225 .
## factor(yearID)2023  13.4136      7.4389  1.803 0.071568 .
## factor(lgID)NL      -5.8805      1.3600 -4.324 1.64e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 25.53 on 1446 degrees of freedom
## (1541 observations deleted due to missingness)
## Multiple R-squared:  0.9514, Adjusted R-squared:  0.9495
## F-statistic: 496.9 on 57 and 1446 DF, p-value: < 2.2e-16

```

```
Teams <- Teams %>% group_by(yearID) %>%
  mutate(
    avgOPS = mean(OPS, na.rm = TRUE),
    avgWHIP = mean(WHIP, na.rm = TRUE),
    avgFP = mean(FP, na.rm = TRUE),
    OPSscale = OPS / avgOPS,
    WHIPscale = avgWHIP / WHIP,
    FPscale = FP / avgFP
  )

model3 <- lm(R ~ OPSscale + WHIPscale + FPscale, data = Teams)
summary(model3)

##
## Call:
## lm(formula = R ~ OPSscale + WHIPscale + FPscale, data = Teams)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -549.99  -28.35    9.85   52.63  200.24
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -463.85     922.20  -0.503  0.61505
## OPSscale      1146.50     106.87  10.728 < 2e-16 ***
## WHIPscale     -263.96      97.87  -2.697  0.00707 **
## FPscale        286.40     908.42   0.315  0.75259
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 93.09 on 1500 degrees of freedom
## (1541 observations deleted due to missingness)
## Multiple R-squared:  0.3304, Adjusted R-squared:  0.329
## F-statistic: 246.7 on 3 and 1500 DF,  p-value: < 2.2e-16
```

Question 2 Choose 3 batters and 3 pitchers that have played in at least 10 seasons and do the following:

- Display the seasonal statistics for these players. The following statistics should be included for batters (derivations of unconventional statistics are in parentheses): year, G, AB, R, H, X2B, X3B, HR, RBI, SB, CS, SBpct (SB / (SB + CS)), BB, SO, OBP, SLG, OPS. The following statistics should be included for pitchers: year, W, L, IPouts, H, ER, HR, BB, HBP, SO, ERA, WHIP, SOper9 (SO / IP * 9), SOperBB (SO / BB). These statistics can be found in or computed from statistics that are found in the **Batting** and **Pitching** dataframes in the **Lahman** package.
- Create career stat lines for each of the players that you selected. Be careful about how these statistics are calculated.
- Provide a plot for career trajectories for one batting and one pitching statistic of your choice. These are two separate graphics, one for the batters and one for the pitchers. The graphics that you produce should display the trajectories of the 3 batters and the 3 pitchers. Provide interesting commentary on your graphic.

```
library(Lahman)
library(dplyr)
library(ggplot2)
```

```

batters <- Batting %>%
  group_by(playerID) %>%
  filter(n_distinct(yearID) >= 10) %>%
  summarise(total_G = sum(G), total_HR = sum(HR)) %>%
  arrange(desc(total_HR)) %>%
  slice(1:3) %>%
  pull(playerID)

pitchers <- Pitching %>%
  group_by(playerID) %>%
  filter(n_distinct(yearID) >= 10) %>%
  summarise(total_W = sum(W), total_SO = sum(SO)) %>%
  arrange(desc(total_SO)) %>%
  slice(1:3) %>%
  pull(playerID)

batting_stats <- Batting %>%
  filter(playerID %in% batters) %>%
  mutate(SBpct = SB / (SB + CS),
         OBP = (H + BB) / (AB + BB),
         SLG = (H + X2B + 2 * X3B + 3 * HR) / AB,
         OPS = OBP + SLG) %>%
  select(playerID, yearID, G, AB, R, H, X2B, X3B, HR, RBI, SB, CS, SBpct, BB, SO, OBP, SLG, OPS)

pitching_stats <- Pitching %>%
  filter(playerID %in% pitchers) %>%
  mutate(ERA = (ER / IPouts) * 27,
         WHIP = (BB + H) / (IPouts / 3),
         SOper9 = (SO / (IPouts / 3)) * 9,
         SOperBB = SO / BB) %>%
  select(playerID, yearID, W, L, IPouts, H, ER, HR, BB, HBP, SO, ERA, WHIP, SOper9, SOperBB)

career_batting <- batting_stats %>%
  group_by(playerID) %>%
  summarise(across(where(is.numeric), sum, na.rm = TRUE))

## Warning: There was 1 warning in `summarise()`.
## i In argument: `across(where(is.numeric), sum, na.rm = TRUE)` .
## i In group 1: `playerID = "aaronha01"` .
## Caused by warning:
## ! The `...` argument of `across()` is deprecated as of dplyr 1.1.0.
## Supply arguments directly to `fns` through an anonymous function instead.
##
## # Previously
## across(a:b, mean, na.rm = TRUE)
##
## # Now
## across(a:b, \(x) mean(x, na.rm = TRUE))

career_pitching <- pitching_stats %>%
  group_by(playerID) %>%
  summarise(across(where(is.numeric), sum, na.rm = TRUE))

batting_plot <- ggplot(batting_stats, aes(x = yearID, y = OPS, color = playerID)) +

```

```

geom_line() +
geom_point() +
labs(title = "OPS Career Trajectory of Selected Batters", x = "Year", y = "OPS") +
theme_minimal()

pitching_plot <- ggplot(pitching_stats, aes(x = yearID, y = S0per9, color = playerID)) +
  geom_line() +
  geom_point() +
  labs(title = "Strikeouts Per 9 Innings Career Trajectory of Selected Pitchers", x = "Year", y = "SO p
  theme_minimal()

print(batting_stats)

```

	playerID	yearID	G	AB	R	H	X2B	X3B	HR	RBI	SB	CS	SBpct	BB	S0
## 1	aaronha01	1954	122	468	58	131	27	6	13	69	2	2	0.5000000	28	39
## 2	aaronha01	1955	153	602	105	189	37	9	27	106	3	1	0.7500000	49	61
## 3	aaronha01	1956	153	609	106	200	34	14	26	92	2	4	0.3333333	37	54
## 4	aaronha01	1957	151	615	118	198	27	6	44	132	1	1	0.5000000	57	58
## 5	aaronha01	1958	153	601	109	196	34	4	30	95	4	1	0.8000000	59	49
## 6	aaronha01	1959	154	629	116	223	46	7	39	123	8	0	1.0000000	51	54
## 7	aaronha01	1960	153	590	102	172	20	11	40	126	16	7	0.6956522	60	63
## 8	aaronha01	1961	155	603	115	197	39	10	34	120	21	9	0.7000000	56	64
## 9	aaronha01	1962	156	592	127	191	28	6	45	128	15	7	0.6818182	66	73
## 10	aaronha01	1963	161	631	121	201	29	4	44	130	31	5	0.8611111	78	94
## 11	aaronha01	1964	145	570	103	187	30	2	24	95	22	4	0.8461538	62	46
## 12	aaronha01	1965	150	570	109	181	40	1	32	89	24	4	0.8571429	60	81
## 13	aaronha01	1966	158	603	117	168	23	1	44	127	21	3	0.8750000	76	96
## 14	aaronha01	1967	155	600	113	184	37	3	39	109	17	6	0.7391304	63	97
## 15	aaronha01	1968	160	606	84	174	33	4	29	86	28	5	0.8484848	64	62
## 16	aaronha01	1969	147	547	100	164	30	3	44	97	9	10	0.4736842	87	47
## 17	aaronha01	1970	150	516	103	154	26	1	38	118	9	0	1.0000000	74	63
## 18	aaronha01	1971	139	495	95	162	22	3	47	118	1	1	0.5000000	71	58
## 19	aaronha01	1972	129	449	75	119	10	0	34	77	4	0	1.0000000	92	55
## 20	aaronha01	1973	120	392	84	118	12	1	40	96	1	1	0.5000000	68	51
## 21	aaronha01	1974	112	340	47	91	16	0	20	69	1	0	1.0000000	39	29
## 22	aaronha01	1975	137	465	45	109	16	2	12	60	0	1	0.0000000	70	51
## 23	aaronha01	1976	85	271	22	62	8	0	10	35	0	1	0.0000000	35	38
## 24	bondsba01	1986	113	413	72	92	26	3	16	48	36	7	0.8372093	65	102
## 25	bondsba01	1987	150	551	99	144	34	9	25	59	32	10	0.7619048	54	88
## 26	bondsba01	1988	144	538	97	152	30	5	24	58	17	11	0.6071429	72	82
## 27	bondsba01	1989	159	580	96	144	34	6	19	58	32	10	0.7619048	93	93
## 28	bondsba01	1990	151	519	104	156	32	3	33	114	52	13	0.8000000	93	83
## 29	bondsba01	1991	153	510	95	149	28	5	25	116	43	13	0.7678571	107	73
## 30	bondsba01	1992	140	473	109	147	36	5	34	103	39	8	0.8297872	127	69
## 31	bondsba01	1993	159	539	129	181	38	4	46	123	29	12	0.7073171	126	79
## 32	bondsba01	1994	112	391	89	122	18	1	37	81	29	9	0.7631579	74	43
## 33	bondsba01	1995	144	506	109	149	30	7	33	104	31	10	0.7560976	120	83
## 34	bondsba01	1996	158	517	122	159	27	3	42	129	40	7	0.8510638	151	76
## 35	bondsba01	1997	159	532	123	155	26	5	40	101	37	8	0.8222222	145	87
## 36	bondsba01	1998	156	552	120	167	44	7	37	122	28	12	0.7000000	130	92
## 37	bondsba01	1999	102	355	91	93	20	2	34	83	15	2	0.8823529	73	62
## 38	bondsba01	2000	143	480	129	147	28	4	49	106	11	3	0.7857143	117	77
## 39	bondsba01	2001	153	476	129	156	32	2	73	137	13	3	0.8125000	177	93
## 40	bondsba01	2002	143	403	117	149	31	2	46	110	9	2	0.8181818	198	47

## 41	bondsba01	2003	130	390	111	133	22	1	45	90	7	0	1.0000000	148	58
## 42	bondsba01	2004	147	373	129	135	27	3	45	101	6	1	0.8571429	232	41
## 43	bondsba01	2005	14	42	8	12	1	0	5	10	0	0	NaN	9	6
## 44	bondsba01	2006	130	367	74	99	23	0	26	77	3	0	1.0000000	115	51
## 45	bondsba01	2007	126	340	75	94	14	0	28	66	5	0	1.0000000	132	54
## 46	ruthba01	1914	5	10	1	2	1	0	0	2	0	NA	NA	0	4
## 47	ruthba01	1915	42	92	16	29	10	1	4	21	0	NA	NA	9	23
## 48	ruthba01	1916	67	136	18	37	5	3	3	15	0	NA	NA	10	23
## 49	ruthba01	1917	52	123	14	40	6	3	2	12	0	NA	NA	12	18
## 50	ruthba01	1918	95	317	50	95	26	11	11	66	6	NA	NA	58	58
## 51	ruthba01	1919	130	432	103	139	34	12	29	114	7	NA	NA	101	58
## 52	ruthba01	1920	142	457	158	172	36	9	54	137	14	14	0.5000000	150	80
## 53	ruthba01	1921	152	540	177	204	44	16	59	171	17	13	0.5666667	145	81
## 54	ruthba01	1922	110	406	94	128	24	8	35	99	2	5	0.2857143	84	80
## 55	ruthba01	1923	152	522	151	205	45	13	41	131	17	21	0.4473684	170	93
## 56	ruthba01	1924	153	529	143	200	39	7	46	121	9	13	0.4090909	142	81
## 57	ruthba01	1925	98	359	61	104	12	2	25	66	2	4	0.3333333	59	68
## 58	ruthba01	1926	152	495	139	184	30	5	47	150	11	9	0.5500000	144	76
## 59	ruthba01	1927	151	540	158	192	29	8	60	164	7	6	0.5384615	137	89
## 60	ruthba01	1928	154	536	163	173	29	8	54	142	4	5	0.4444444	137	87
## 61	ruthba01	1929	135	499	121	172	26	6	46	154	5	3	0.6250000	72	60
## 62	ruthba01	1930	145	518	150	186	28	9	49	153	10	10	0.5000000	136	61
## 63	ruthba01	1931	145	534	149	199	31	3	46	163	5	4	0.5555556	128	51
## 64	ruthba01	1932	133	457	120	156	13	5	41	137	2	2	0.5000000	130	62
## 65	ruthba01	1933	137	459	97	138	21	3	34	103	4	5	0.4444444	114	90
## 66	ruthba01	1934	125	365	78	105	17	4	22	84	1	3	0.2500000	104	63
## 67	ruthba01	1935	28	72	13	13	0	0	6	12	0	NA	NA	20	24
##	OBP	SLG	OPS												
## 1	0.3205645	0.4465812	0.7671457												
## 2	0.3655914	0.5398671	0.9054585												
## 3	0.3668731	0.5582923	0.9251653												
## 4	0.3794643	0.6000000	0.9794643												
## 5	0.3863636	0.5457571	0.9321207												
## 6	0.4029412	0.6359300	1.0388712												
## 7	0.3569231	0.5661017	0.9230248												
## 8	0.3839150	0.5936982	0.9776132												
## 9	0.3905775	0.6182432	1.0088208												
## 10	0.3935120	0.5863708	0.9798828												
## 11	0.3939873	0.5140351	0.9080224												
## 12	0.3825397	0.5596491	0.9421888												
## 13	0.3593520	0.5389718	0.8983238												
## 14	0.3725490	0.5733333	0.9458824												
## 15	0.3552239	0.4983498	0.8535737												
## 16	0.3958991	0.6069470	1.0028460												
## 17	0.3864407	0.5736434	0.9600841												
## 18	0.4116608	0.6686869	1.0803476												
## 19	0.3900185	0.5144766	0.9044951												
## 20	0.4043478	0.6428571	1.0472050												
## 21	0.3430079	0.4911765	0.8341844												
## 22	0.3345794	0.3548387	0.6894181												
## 23	0.3169935	0.3690037	0.6859972												
## 24	0.3284519	0.4164649	0.7449168												
## 25	0.3272727	0.4918330	0.8191058												
## 26	0.3672131	0.4907063	0.8579194												

```
## 27 0.3521545 0.4258621 0.7780166
## 28 0.4068627 0.5645472 0.9714100
## 29 0.4149109 0.5137255 0.9286363
## 30 0.4566667 0.6236786 1.0803453
## 31 0.4616541 0.6771800 1.1388341
## 32 0.4215054 0.6470588 1.0685642
## 33 0.4297125 0.5770751 1.0067876
## 34 0.4640719 0.6150870 1.0791589
## 35 0.4431315 0.5845865 1.0277179
## 36 0.4354839 0.6086957 1.0441795
## 37 0.3878505 0.6169014 1.0047519
## 38 0.4422111 0.6875000 1.1297111
## 39 0.5099541 0.8634454 1.3733994
## 40 0.5773710 0.7990074 1.3763785
## 41 0.5223048 0.7487179 1.2710228
## 42 0.6066116 0.8123324 1.4189440
## 43 0.4117647 0.6666667 1.0784314
## 44 0.4439834 0.5449591 0.9889425
## 45 0.4788136 0.5647059 1.0435194
## 46 0.2000000 0.3000000 0.5000000
## 47 0.3762376 0.5760870 0.9523246
## 48 0.3219178 0.4191176 0.7410355
## 49 0.3851852 0.4715447 0.8567299
## 50 0.4080000 0.5552050 0.9632050
## 51 0.4502814 0.6574074 1.1076888
## 52 0.5304778 0.8490153 1.3794931
## 53 0.5094891 0.8462963 1.3557853
## 54 0.4326531 0.6724138 1.1050669
## 55 0.5419075 0.7643678 1.3062753
## 56 0.5096870 0.7391304 1.2488175
## 57 0.3899522 0.5431755 0.9331276
## 58 0.5133020 0.7373737 1.2506758
## 59 0.4859675 0.7722222 1.2581897
## 60 0.4606241 0.7089552 1.1695793
## 61 0.4273205 0.6973948 1.1247153
## 62 0.4923547 0.7316602 1.2240150
## 63 0.4939577 0.7003745 1.1943322
## 64 0.4872232 0.6608315 1.1480547
## 65 0.4397906 0.5816993 1.0214899
## 66 0.4456290 0.5369863 0.9826153
## 67 0.3586957 0.4305556 0.7892512
```

```
print(pitching_stats)
```

```
##      playerID yearID  W  L IPouts   H  ER HR  BB HBP  SO      ERA      WHIP
## 1  clemereo02  1984   9  4   400 146  64 13  29   2 126  4.320000 1.3125000
## 2  clemereo02  1985   7  5   295  83  36  5  37   3  74  3.294915 1.2203390
## 3  clemereo02  1986  24  4   762 179  70 21  67   4 238  2.480315 0.9685039
## 4  clemereo02  1987  20  9   845 248  93 19  83   9 256  2.971598 1.1751479
## 5  clemereo02  1988  18 12   792 217  86 17  62   6 291  2.931818 1.0568182
## 6  clemereo02  1989  17 11   760 215  88 20  93   8 230  3.126316 1.2157895
## 7  clemereo02  1990  21  6   685 193  49  7  54   7 209  1.931387 1.0817518
## 8  clemereo02  1991  18 10   814 219  79 15  65   5 241  2.620393 1.0466830
## 9  clemereo02  1992  18 11   740 203  66 11  62   9 208  2.408108 1.0743243
## 10 clemereo02  1993  11 14   575 175  95 17  67  11 160  4.460870 1.2626087
```

## 11	clemero02	1994	9	7	512	124	54	15	71	4	168	2.847656	1.1425781
## 12	clemero02	1995	10	5	420	141	65	15	60	14	132	4.178571	1.4357143
## 13	clemero02	1996	10	13	728	216	98	19	106	4	257	3.634615	1.3269231
## 14	clemero02	1997	21	7	792	204	60	9	68	12	292	2.045455	1.0303030
## 15	clemero02	1998	20	6	704	169	69	11	88	7	271	2.646307	1.0951705
## 16	clemero02	1999	14	10	563	185	96	20	90	9	163	4.603908	1.4653641
## 17	clemero02	2000	13	8	613	184	84	26	84	10	188	3.699837	1.3115824
## 18	clemero02	2001	20	3	661	205	86	19	72	5	213	3.512859	1.2571861
## 19	clemero02	2002	13	6	540	172	87	18	63	7	192	4.350000	1.3055556
## 20	clemero02	2003	17	9	635	199	92	24	58	5	190	3.911811	1.2141732
## 21	clemero02	2004	18	4	643	169	71	15	79	6	218	2.981337	1.1570762
## 22	clemero02	2005	13	8	634	151	44	11	62	3	185	1.873817	1.0078864
## 23	clemero02	2006	7	6	340	89	29	7	29	4	102	2.302941	1.0411765
## 24	clemero02	2007	6	6	297	99	46	9	31	5	68	4.181818	1.3131313
## 25	johnsra05	1988	3	0	78	23	7	3	7	0	25	2.423077	1.1538462
## 26	johnsra05	1989	0	4	89	29	22	2	26	0	26	6.674157	1.8539326
## 27	johnsra05	1989	7	9	393	118	64	11	70	3	104	4.396947	1.4351145
## 28	johnsra05	1990	14	11	659	174	89	26	120	5	194	3.646434	1.3383915
## 29	johnsra05	1991	13	10	604	151	89	15	152	12	228	3.978477	1.5049669
## 30	johnsra05	1992	12	14	631	154	88	13	144	18	241	3.765452	1.4167987
## 31	johnsra05	1993	19	8	766	185	92	22	99	16	308	3.242820	1.1122715
## 32	johnsra05	1994	13	6	516	132	61	14	72	6	204	3.191860	1.1860465
## 33	johnsra05	1995	18	2	643	159	59	12	65	6	294	2.477449	1.0451011
## 34	johnsra05	1996	5	0	184	48	25	8	25	2	85	3.668478	1.1902174
## 35	johnsra05	1997	20	4	639	147	54	20	77	10	291	2.281690	1.0516432
## 36	johnsra05	1998	9	10	480	146	77	19	60	11	213	4.331250	1.2875000
## 37	johnsra05	1998	10	1	253	57	12	4	26	3	116	1.280632	0.9841897
## 38	johnsra05	1999	17	9	815	207	75	30	70	9	364	2.484663	1.0196319
## 39	johnsra05	2000	19	7	746	202	73	23	76	6	347	2.642091	1.1179625
## 40	johnsra05	2001	21	6	749	181	69	19	71	18	372	2.487316	1.0093458
## 41	johnsra05	2002	24	5	780	197	67	26	71	13	334	2.319231	1.0307692
## 42	johnsra05	2003	6	8	342	125	54	16	27	8	125	4.263158	1.3333333
## 43	johnsra05	2004	16	14	737	177	71	18	44	10	290	2.601085	0.8995929
## 44	johnsra05	2005	17	8	677	207	95	32	47	12	211	3.788774	1.1255539
## 45	johnsra05	2006	17	11	615	194	114	28	60	10	172	5.004878	1.2390244
## 46	johnsra05	2007	4	3	170	52	24	7	13	4	72	3.811765	1.1470588
## 47	johnsra05	2008	11	10	552	184	80	24	44	6	173	3.913043	1.2391304
## 48	johnsra05	2009	8	6	288	97	52	19	31	2	86	4.875000	1.3333333
## 49	ryananno01	1966	0	1	9	5	5	1	3	0	6	15.000000	2.6666667
## 50	ryananno01	1968	6	9	402	93	46	12	75	4	133	3.089552	1.2537313
## 51	ryananno01	1969	6	3	268	60	35	3	53	1	92	3.526119	1.2649254
## 52	ryananno01	1970	7	11	395	86	50	10	97	4	125	3.417722	1.3898734
## 53	ryananno01	1971	10	14	456	125	67	8	116	15	137	3.967105	1.5855263
## 54	ryananno01	1972	19	16	852	166	72	14	157	10	329	2.281690	1.1373239
## 55	ryananno01	1973	21	16	978	238	104	18	162	7	383	2.871166	1.2269939
## 56	ryananno01	1974	22	16	998	221	107	18	202	9	367	2.894790	1.2715431
## 57	ryananno01	1975	14	12	594	152	76	13	132	7	186	3.454545	1.4343434
## 58	ryananno01	1976	17	18	853	193	106	13	183	5	327	3.355217	1.3223916
## 59	ryananno01	1977	19	16	897	198	92	12	204	9	341	2.769231	1.3444816
## 60	ryananno01	1978	10	13	704	183	97	12	148	3	260	3.720170	1.4105114
## 61	ryananno01	1979	16	14	668	169	89	15	114	6	223	3.597305	1.2709581
## 62	ryananno01	1980	11	10	701	205	87	10	98	3	200	3.350927	1.2967190
## 63	ryananno01	1981	11	5	447	99	28	2	68	1	140	1.691275	1.1208054
## 64	ryananno01	1982	16	12	751	196	88	20	109	8	245	3.163782	1.2183755

## 65	ryanno01	1983	14	9	589	134	65	9	101	4	183	2.979626	1.1969440
## 66	ryanno01	1984	12	11	551	143	62	12	69	4	197	3.038113	1.1542650
## 67	ryanno01	1985	10	12	696	205	98	12	95	9	209	3.801724	1.2931034
## 68	ryanno01	1986	12	8	534	119	66	14	82	4	194	3.337079	1.1292135
## 69	ryanno01	1987	8	16	635	154	65	14	87	4	270	2.763780	1.1385827
## 70	ryanno01	1988	12	11	660	186	86	18	87	7	228	3.518182	1.2409091
## 71	ryanno01	1989	16	10	718	162	85	17	98	9	301	3.196379	1.0863510
## 72	ryanno01	1990	13	9	612	137	78	18	74	7	232	3.441176	1.0343137
## 73	ryanno01	1991	12	6	519	102	56	12	72	5	203	2.913295	1.0057803
## 74	ryanno01	1992	5	9	472	138	65	9	69	12	157	3.718220	1.3156780
## 75	ryanno01	1993	5	5	199	54	36	5	40	1	46	4.884422	1.4170854
##	S0per9	S0perBB											
## 1	8.505000	4.344828											
## 2	6.772881	2.000000											
## 3	8.433071	3.552239											
## 4	8.179882	3.084337											
## 5	9.920455	4.693548											
## 6	8.171053	2.473118											
## 7	8.237956	3.870370											
## 8	7.993857	3.707692											
## 9	7.589189	3.354839											
## 10	7.513043	2.388060											
## 11	8.859375	2.366197											
## 12	8.485714	2.200000											
## 13	9.531593	2.424528											
## 14	9.954545	4.294118											
## 15	10.393466	3.079545											
## 16	7.817052	1.811111											
## 17	8.280587	2.238095											
## 18	8.700454	2.958333											
## 19	9.600000	3.047619											
## 20	8.078740	3.275862											
## 21	9.153966	2.759494											
## 22	7.878549	2.983871											
## 23	8.100000	3.517241											
## 24	6.181818	2.193548											
## 25	8.653846	3.571429											
## 26	7.887640	1.000000											
## 27	7.145038	1.485714											
## 28	7.948407	1.616667											
## 29	10.192053	1.500000											
## 30	10.312203	1.673611											
## 31	10.856397	3.111111											
## 32	10.674419	2.833333											
## 33	12.345257	4.523077											
## 34	12.472826	3.400000											
## 35	12.295775	3.779221											
## 36	11.981250	3.550000											
## 37	12.379447	4.461538											
## 38	12.058896	5.200000											
## 39	12.558981	4.565789											
## 40	13.409880	5.239437											
## 41	11.561538	4.704225											
## 42	9.868421	4.629630											

```
## 43 10.624152 6.590909
## 44 8.415066 4.489362
## 45 7.551220 2.866667
## 46 11.435294 5.538462
## 47 8.461957 3.931818
## 48 8.062500 2.774194
## 49 18.000000 2.000000
## 50 8.932836 1.773333
## 51 9.268657 1.735849
## 52 8.544304 1.288660
## 53 8.111842 1.181034
## 54 10.426056 2.095541
## 55 10.573620 2.364198
## 56 9.928858 1.816832
## 57 8.454545 1.409091
## 58 10.350528 1.786885
## 59 10.264214 1.671569
## 60 9.971591 1.756757
## 61 9.013473 1.956140
## 62 7.703281 2.040816
## 63 8.456376 2.058824
## 64 8.808256 2.247706
## 65 8.388795 1.811881
## 66 9.653358 2.855072
## 67 8.107759 2.200000
## 68 9.808989 2.365854
## 69 11.480315 3.103448
## 70 9.327273 2.620690
## 71 11.318942 3.071429
## 72 10.235294 3.135135
## 73 10.560694 2.819444
## 74 8.980932 2.275362
## 75 6.241206 1.150000
```

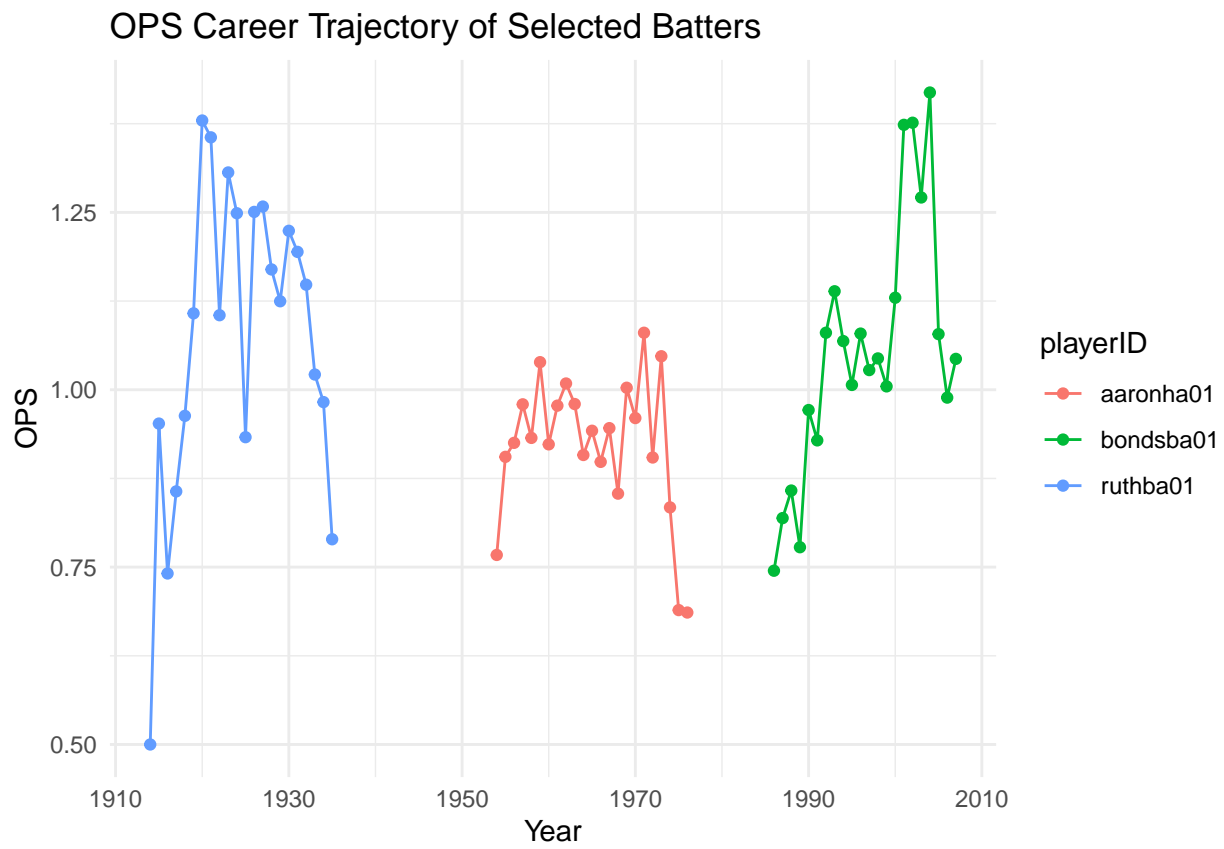
```
print(career_batting)
```

```
## # A tibble: 3 x 18
##   playerID yearID      G      AB      R      H      X2B      X3B      HR      RBI      SB      CS
##   <chr>      <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int>
## 1 aaronha01  45195  3298 12364  2174  3771   624    98   755  2297   240    73
## 2 bondsba01  43923  2986  9847  2227  2935   601    77   762  1996   514   141
## 3 ruthba01   42339  2503  8398  2174  2873   506   136   714  2217   123   117
## # i 6 more variables: SBpct <dbl>, BB <int>, SO <int>, OBP <dbl>, SLG <dbl>,
## #   OPS <dbl>
```

```
print(career_pitching)
```

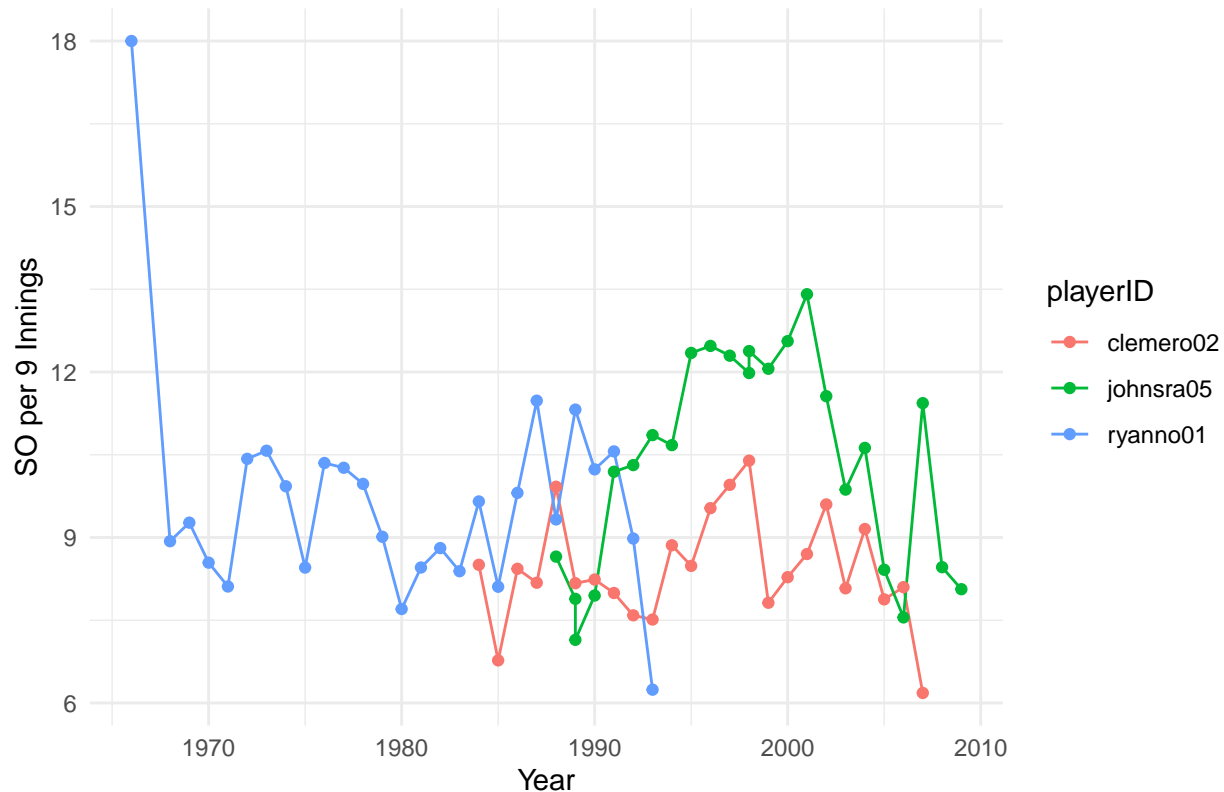
```
## # A tibble: 3 x 15
##   playerID yearID      W      L IPouts      H      ER      HR      BB      HBP      SO      ERA
##   <chr>      <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <dbl>
## 1 clemereo02  47892   354   184 14750  4185  1707   363  1580   159  4672  77.3
## 2 johnsra05  47954   303   166 12406  3346  1513   411  1497   190  4875  83.5
## 3 ryanno01   53459   324   292 16158  3923  1911   321  2795   158  5714  99.7
## # i 3 more variables: WHIP <dbl>, SOper9 <dbl>, SOperBB <dbl>
```

```
print(batting_plot)
```



```
print(pitching_plot)
```

Strikeouts Per 9 Innings Career Trajectory of Selected Pitchers



Question 3 Exercise 1.2 in the online version of Analyzing Baseball Data with R. Exercise 2 on page 27 of the textbook.

```
gibson_1968 <- Pitching %>%
  filter(playerID == "gibsobo01", yearID == 1968)

fraction_completed <- gibson_1968$CG / gibson_1968$GS
strikeout_to_walk_ratio <- gibson_1968$SO / gibson_1968$BB
innings_pitched <- gibson_1968$IPouts / 3
WHIP <- (gibson_1968$BB + gibson_1968$H) / innings_pitched

list(
  fraction_completed = fraction_completed,
  strikeout_to_walk_ratio = strikeout_to_walk_ratio,
  innings_pitched = innings_pitched,
  WHIP = WHIP
)

## $fraction_completed
## [1] 0.8235294
##
## $strikeout_to_walk_ratio
## [1] 4.322581
##
## $innings_pitched
## [1] 304.6667
##
```

```
## $WHIP
## [1] 0.8533917
```

Question 4 Exercise 1.3 in the online version of Analyzing Baseball Data with R. Exercise 3 on page 27 of the textbook.

```
library(readr)
library(dplyr)

file_path <- "/Users/Rhea/Desktop/stat430/sp25_stat430_rheaaa2/rhea_labs/lab1/gl1964.txt"

column_names <- c(
  "Date", "DoubleHeader", "DayOfWeek", "VisitingTeam", "VisitingTeamLeague",
  "VisitingTeamGameNumber", "HomeTeam", "HomeTeamLeague", "HomeTeamGameNumber",
  "VisitorRunsScored", "HomeRunsScore", "LengthInOuts", "DayNight",
  "CompletionInfo", "ForfeitInfo", "ProtestInfo", "ParkID", "Attendance",
  "Duration", "VisitorLineScore", "HomeLineScore", "VisitorAB", "VisitorH",
  "VisitorD", "VisitorT", "VisitorHR", "VisitorRBI", "VisitorSH", "VisitorSF",
  "VisitorHBP", "VisitorBB", "VisitorIBB", "VisitorK", "VisitorSB",
  "VisitorCS", "VisitorGDP", "VisitorCI", "VisitorLOB"
)

game_logs <- read_csv(file_path, col_names = column_names)

## Warning: One or more parsing issues, call `problems()` on your data frame for details,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)

## Rows: 1626 Columns: 161
## -- Column specification -----
## Delimiter: ","
## chr (74): DayOfWeek, VisitingTeam, VisitingTeamLeague, HomeTeam, HomeTeamLea...
## dbl (83): Date, DoubleHeader, VisitingTeamGameNumber, HomeTeamGameNumber, Vi...
## lgl (4): CompletionInfo, ForfeitInfo, X86, X88
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

game <- game_logs %>%
  filter(Date == "19640621", VisitingTeam == "PHI", HomeTeam == "NYN")

game_duration <- as.numeric(game$Duration)
game_hours <- game_duration %/% 60
game_minutes <- game_duration %% 60
cat("Game Duration:", game_hours, "hours", game_minutes, "minutes\n")

## Game Duration: 2 2 hours 19 51 minutes

cat("The attendance value is likely zero due to missing or unrecorded data in the log.\n")

## The attendance value is likely zero due to missing or unrecorded data in the log.

extra_base_hits <- as.numeric(game$VisitorD) + as.numeric(game$VisitorT) + as.numeric(game$VisitorHR)
cat("Phillies' Extra-Base Hits:", extra_base_hits, "\n")

## Phillies' Extra-Base Hits: 3 4
```



```

visitor_hits <- as.numeric(game$VisitorH)
visitor_walks <- as.numeric(game$VisitorBB)
visitor_hit_by_pitch <- as.numeric(game$VisitorHBP)
visitor_at_bats <- as.numeric(game$VisitorAB)
visitor_sacrifice_flies <- as.numeric(game$VisitorSF)

OBP <- (visitor_hits + visitor_walks + visitor_hit_by_pitch) / (visitor_at_bats + visitor_walks + visitor_sacrifice_flies)
cat("Phillies' On-Base Percentage:", round(OBP, 3), "\n")

## Phillies' On-Base Percentage: 0.333 0.381

```