

MATH 630 Midterm

Joshua Burkhardt

Midterm: Simple Linear Regression

Overview

```
Teams <- Lahman::Teams
Salaries <- Lahman::Salaries
```

HLO Lahman

```
str(Teams)
str(Salaries)
glimpse(Teams)
glimpse(Salaries)
head(Teams)
head(Salaries)
tail(Teams)
tail(Salaries)
names(Teams)
names(Salaries)
ncol(Teams)
ncol(Salaries)
length(Teams)
length(Salaries)
head(rownames(Teams))
head(rownames(Salaries))
dim(Teams)
dim(Salaries)
nrow(Teams)
nrow(Salaries)
summary(Teams)
summary(Salaries)
?Teams
?Salaries
```

Are they data.frames, matrices, vectors, lists?

data.frames

What is the unit of analysis in the dataset?

Teams: One row per team-year.

Salaries: One row per player-team-year.

How many variables/columns?

Teams: 48
Salaries: 5

How many rows/observations?

Teams: 2775
Salaries: 24758

Find the variables for games won, team, year, and salary.

W: Wins
teamID: Team; a factor
yearID: Year
salary: Salary

Which variables are continuous?

In theory, salary could be continuous but in practice, salary looks like it's rounded to the nearest thousand.

Which variables are discrete?

W
teamID
yearID
salary

Which variables are categorical?

teamID

How many levels do they have?

149

What about missing data for any variables?

```
unique(is.na(Teams))  
unique(is.na(Salaries))
```

Teams: Missing data in several columns
Salaries: No missing data reported

Data wrangling in dplyr

Create a new dataset that includes total yearly payroll for each team in the Salaries dataframe.

```
typ <- Salaries %>% group_by(yearID,teamID) %>% summarise payroll = sum(salary))
head(typ)
```

Source: local data frame [6 x 3]
Groups: yearID [1]

```
  yearID teamID payroll
    (int) (fctr)   (int)
1  1985    ATL 14807000
2  1985    BAL 11560712
3  1985    BOS 10897560
4  1985    CAL 14427894
5  1985    CHA  9846178
6  1985    CHN 12702917
```

Add this payroll column to the Teams dataframe.

```
teams_pay <- inner_join(Teams,typ,c("yearID","teamID"))
head(teams_pay)
```

```
  yearID lgID teamID franchID divID Rank   G Ghome W  L DivWin WCWin
1  1985  NL   ATL     ATL     W    5 162   81 66 96     N <NA>
2  1985  AL   BAL     BAL     E    4 161   81 83 78     N <NA>
3  1985  AL   BOS     BOS     E    5 163   81 81 81     N <NA>
4  1985  AL   CAL     ANA     W    2 162   79 90 72     N <NA>
5  1985  AL   CHA     CHW     W    3 163   81 85 77     N <NA>
6  1985  NL   CHN     CHC     E    4 162   81 77 84     N <NA>
  LgWin WSWin  R  AB   H X2B X3B  HR  BB  SO  SB CS HBP SF  RA  ER  ERA
1     N     N 632 5526 1359 213 28 126 553 849 72 52 NA NA 781 678 4.19
2     N     N 818 5517 1451 234 22 214 604 908 69 43 NA NA 764 694 4.38
3     N     N 800 5720 1615 292 31 162 562 816 66 27 NA NA 720 659 4.06
4     N     N 732 5442 1364 215 31 153 648 902 106 51 NA NA 703 633 3.91
5     N     N 736 5470 1386 247 37 146 471 843 108 56 NA NA 720 656 4.07
6     N     N 686 5492 1397 239 28 150 562 937 182 49 NA NA 729 667 4.16
  CG SHO SV IPouts  HA HRA BBA  SOA  E  DP  FP           name
1  9   9 29  4371 1512 134 642 776 159 197 0.97   Atlanta Braves
2 32   6 33  4281 1480 160 568 793 115 168 0.98 Baltimore Orioles
3 35   8 29  4383 1487 130 540 913 145 161 0.97   Boston Red Sox
4 22   8 41  4371 1453 171 514 767 112 202 0.98 California Angels
5 20   8 39  4353 1411 161 569 1023 111 152 0.98 Chicago White Sox
6 20   8 42  4326 1492 156 519 820 134 150 0.97   Chicago Cubs
  park attendance BPF PPF teamIDBR teamIDlahman45
1 Atlanta-Fulton County Stadium 1350137 105 106    ATL    ATL
2      Memorial Stadium 2132387 97 97    BAL    BAL
3      Fenway Park II 1786633 104 104    BOS    BOS
4      Anaheim Stadium 2567427 100 100    CAL    CAL
5      Comiskey Park 1669888 104 104    CHW    CHA
6      Wrigley Field 2161534 110 110    CHC    CHN
  teamIDretro payroll
1      ATL 14807000
2      BAL 11560712
3      BOS 10897560
```

```

4      CAL 14427894
5      CHA 9846178
6      CHN 12702917

```

We'll focus on the years 2000 - 2014. Use dplyr to filter() the dataset you created with the Teams data plus the payroll column for just those years.

```
recent_tpay <- filter(teams_pay, yearID >= 2000, yearID <= 2014)
```

Gift

```

bat_stats <-
  battingStats(data = Lahman::Batting,
               idvars = c("playerID",
                           "yearID",
                           "stint",
                           "teamID",
                           "lgID"), cbind = TRUE)

```

Write a dplyr expression to create a new dataframe that contains means for each of these three new variables for each team and year from 2000 - 2014 (rather than for each player).

```

bat_avgs <- bat_stats %>%
  filter(yearID >= 2000, yearID <= 2014) %>%
  group_by(yearID, teamID) %>%
  summarise(ob_perc = mean(OBP, na.rm = TRUE),
            slug_perc = mean(SlugPct, na.rm = TRUE),
            ops = mean(OPS, na.rm = TRUE))
head(bat_avgs)

```

Source: local data frame [6 x 5]

Groups: yearID [1]

	yearID	teamID	ob_perc	slug_perc	ops
	(int)	(fctr)	(dbl)	(dbl)	(dbl)
1	2000	ANA	0.3385926	0.3756667	0.7142593
2	2000	ARI	0.3016216	0.3618378	0.6634595
3	2000	ATL	0.2430000	0.3009167	0.5439167
4	2000	BAL	0.2474000	0.3088571	0.5562571
5	2000	BOS	0.2644571	0.3063429	0.5708000
6	2000	CHA	0.2937600	0.3396800	0.6334400

Adds these new batting statistic columns to your current dataframe

```

# warning seems ok based on http://goo.gl/9QH3fo
teams_bat <- inner_join(bat_avgs, recent_tpay, c("yearID", "teamID"))
head(teams_bat)

```

Source: local data frame [6 x 52]

Groups: yearID [1]

	yearID	teamID	ob_perc	slug_perc	ops	lgID	franchID	divID	Rank
	(int)	(chr)	(dbl)	(dbl)	(dbl)	(fctr)	(fctr)	(chr)	(int)
1	2000	ANA	0.3385926	0.3756667	0.7142593	AL	ANA	W	3
2	2000	ARI	0.3016216	0.3618378	0.6634595	NL	ARI	W	3
3	2000	ATL	0.2430000	0.3009167	0.5439167	NL	ATL	E	1
4	2000	BAL	0.2474000	0.3088571	0.5562571	AL	BAL	E	4
5	2000	BOS	0.2644571	0.3063429	0.5708000	AL	BOS	E	2
6	2000	CHA	0.2937600	0.3396800	0.6334400	AL	CHW	C	1

Variables not shown: G (int), Ghome (int), W (int), L (int), DivWin (chr), WCWin (chr), LgWin (chr), WSWin (chr), R (int), AB (int), H (int), X2B (int), X3B (int), HR (int), BB (int), SO (int), SB (int), CS (int), HBP (int), SF (int), RA (int), ER (int), ERA (dbl), CG (int), SHO (int), SV (int), IPouts (int), HA (int), HRA (int), BBA (int), SOA (int), E (int), DP (int), FP (dbl), name (chr), park (chr), attendance (int), BPF (int), PPF (int), teamIDBR (chr), teamIDlahman45 (chr), teamIDretro (chr), payroll (int)

Univariate EDA (+ more wrangling)

```
teams_bat %>%
  group_by(teamID) %>%
  tally() %>% arrange(n) %>%
  print(n = 33)
```

Source: local data frame [33 x 2]

	teamID	n
	(chr)	(int)
1	MIA	3
2	ANA	5
3	MON	5
4	LAA	10
5	WAS	10
6	FLO	12
7	ARI	15
8	ATL	15
9	BAL	15
10	BOS	15
11	CHA	15
12	CHN	15
13	CIN	15
14	CLE	15
15	COL	15
16	DET	15
17	HOU	15
18	KCA	15
19	LAN	15
20	MIL	15
21	MIN	15
22	NYA	15
23	NYN	15
24	OAK	15

25	PHI	15
26	PIT	15
27	SDN	15
28	SEA	15
29	SFN	15
30	SLN	15
31	TBA	15
32	TEX	15
33	TOR	15

How many teams are there?

33

Which teams have data for the least number of seasons?

MIA, ANA, and MON

Which have the most seasons?

ARI,ATL,BAL,BOS,CHA,CHN,CIN,CLE,COL,
DET,HOU,KCA,LAN,MIL,MIN,NYA,NYN,OAK,
PHI,PIT,SDN,SEA,SFN,SLN,TBA,TEX,TOR all have 15

```
teams_bat <- teams_bat %>%
  filter(!(teamID %in% c("ANA", "MIA", "MON"))) # you should understand what this does
```

```
teams_bat %>%
  group_by(yearID) %>%
  select(G) %>%
  summarise_each(funs(min,max,mean,median))
```

Source: local data frame [15 x 5]

	yearID (int)	min (int)	max (int)	mean (dbl)	median (dbl)
1	2000	161	163	161.9286	162
2	2001	161	162	161.9286	162
3	2002	161	162	161.7143	162
4	2003	161	163	162.0000	162
5	2004	161	162	161.8571	162
6	2005	162	163	162.0667	162
7	2006	161	162	161.9333	162
8	2007	162	163	162.0667	162
9	2008	161	163	161.8667	162
10	2009	161	163	162.0000	162
11	2010	162	162	162.0000	162
12	2011	161	162	161.9333	162
13	2012	162	162	162.0000	162
14	2013	162	163	162.0690	162
15	2014	162	162	162.0000	162

Is there a lot of variability in number of games played per season across teams?

No

What is the range of games played by teams per season?

2 (163 - 161)

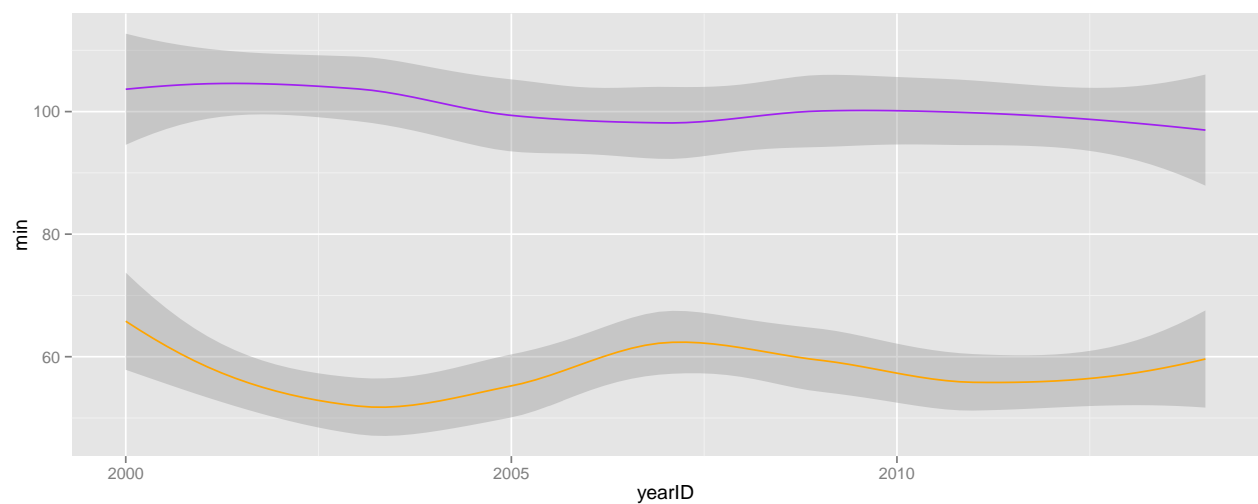
Number of games won

```
teams_bat %>%  
  group_by(yearID) %>%  
  select(W) %>%  
  summarise_each(funs(min,max)) %>% head()
```

Source: local data frame [6 x 3]

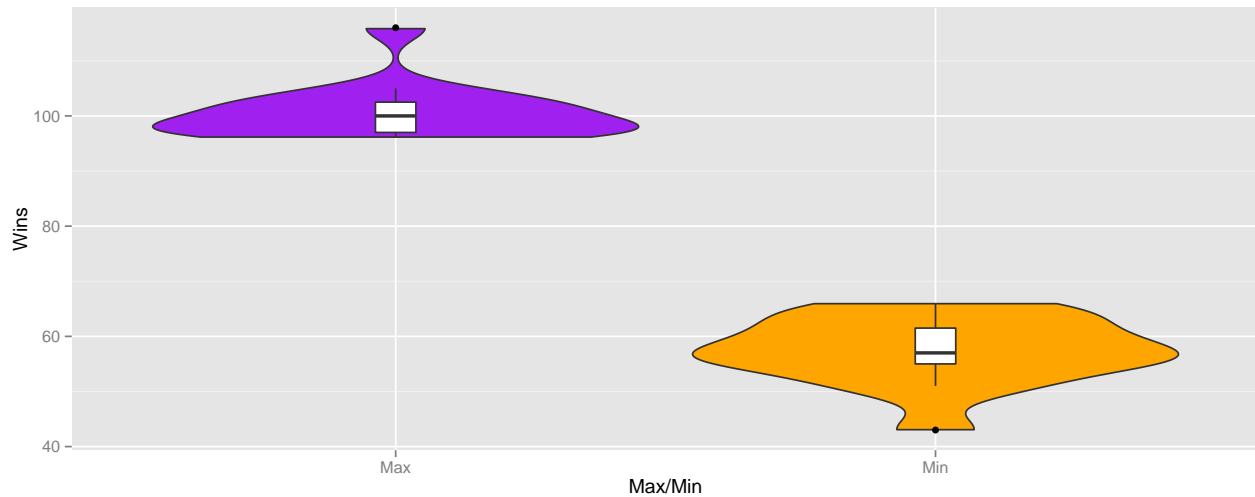
	yearID (int)	min (int)	max (int)
1	2000	65	97
2	2001	62	116
3	2002	55	103
4	2003	43	101
5	2004	51	105
6	2005	56	100

```
teams_bat %>%  
  group_by(yearID) %>%  
  select(W) %>%  
  summarise_each(funs(min,max)) %>% ggplot() +  
  geom_smooth(aes(x=yearID,y=min),color="orange") +  
  geom_smooth(aes(x=yearID,y=max),color="purple")
```



In the figure above, we see that the maximum and minimum number of wins slightly decreases and increases, respectively. This shows that, as a whole, the league is becoming more heterogeneous and competitive.

```
teams_bat %>%
  group_by(yearID) %>%
  select(W) %>%
  summarise_each(funs(min,max)) %>% ggplot() +
  geom_violin(aes(x=factor("Min"),y=min),fill="orange") +
  geom_violin(aes(x=factor("Max"),y=max),fill="purple") +
  geom_boxplot(aes(x=factor("Min"),y=min,width=0.1) +
  geom_boxplot(aes(x=factor("Max"),y=max),width=0.1) +
  ylab("Wins") +
  xlab("Max/Min")
```



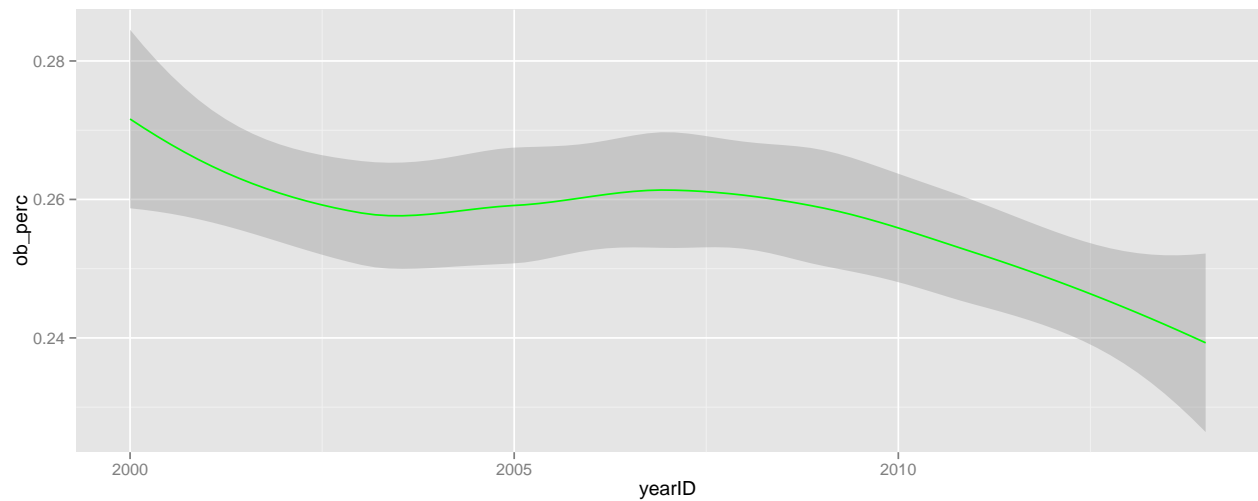
Mean on-base percentage

```
teams_bat %>%
  group_by(yearID) %>%
  select(ob_perc) %>%
  summarise_each(funs(mean)) %>% head()
```

Source: local data frame [6 x 2]

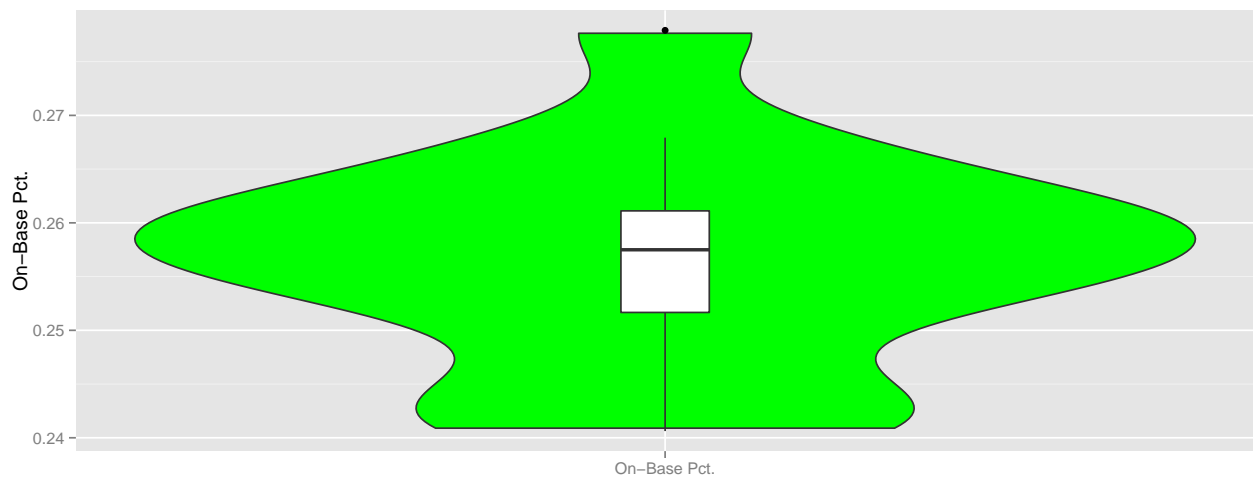
	yearID	ob_perc
	(int)	(dbl)
1	2000	0.2779122
2	2001	0.2554038
3	2002	0.2574950
4	2003	0.2679267
5	2004	0.2494640
6	2005	0.2587080

```
teams_bat %>%
  group_by(yearID) %>%
  select(ob_perc,slug_perc) %>%
  summarise_each(funs(mean)) %>% ggplot() +
  geom_smooth(aes(x=yearID,y=ob_perc),color="green")
```

In the plot above we see that on-base percentage has decreased slightly in recent years.

```
teams_bat %>%
  group_by(yearID) %>%
  select(ob_perc,slug_perc) %>%
  summarise_each(funs(mean)) %>% ggplot() +
  geom_violin(aes(x=factor("On-Base Pct."),y=ob_perc),fill="green") +
  geom_boxplot(aes(x=factor("On-Base Pct."),y=ob_perc),width=0.1) +
  ylab("On-Base Pct.") +
  xlab("")
```



Mean slugging percentage

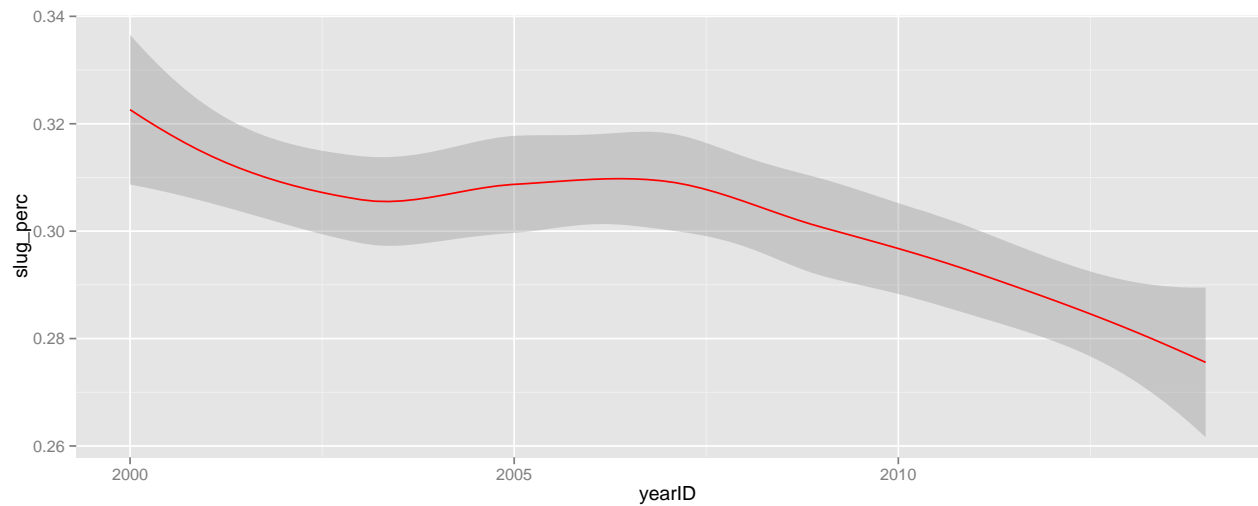
```
teams_bat %>%
  group_by(yearID) %>%
  select(slug_perc) %>%
  summarise_each(funs(mean)) %>% head()
```

Source: local data frame [6 x 2]

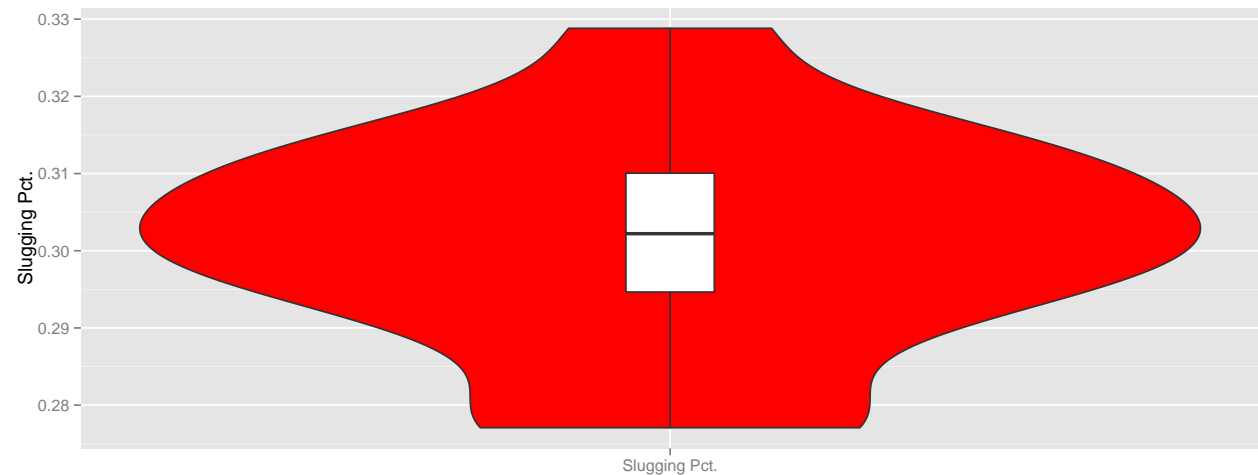
```
yearID slug_perc
```

	(int)	(dbl)
1	2000	0.3288996
2	2001	0.3080489
3	2002	0.3008938
4	2003	0.3138558
5	2004	0.2970384
6	2005	0.3076036

```
teams_bat %>%
  group_by(yearID) %>%
  select(ob_perc,slug_perc) %>%
  summarise_each(funs(mean)) %>% ggplot() +
  geom_smooth(aes(x=yearID,y=slug_perc),color="red")
```



```
teams_bat %>%
  group_by(yearID) %>%
  select(ob_perc,slug_perc) %>%
  summarise_each(funs(mean)) %>% ggplot() +
  geom_violin(aes(x=factor("Slugging Pct."),y=slug_perc),fill="red") +
  geom_boxplot(aes(x=factor("Slugging Pct."),y=slug_perc),width=0.1) +
  ylab("Slugging Pct.") +
  xlab("")
```



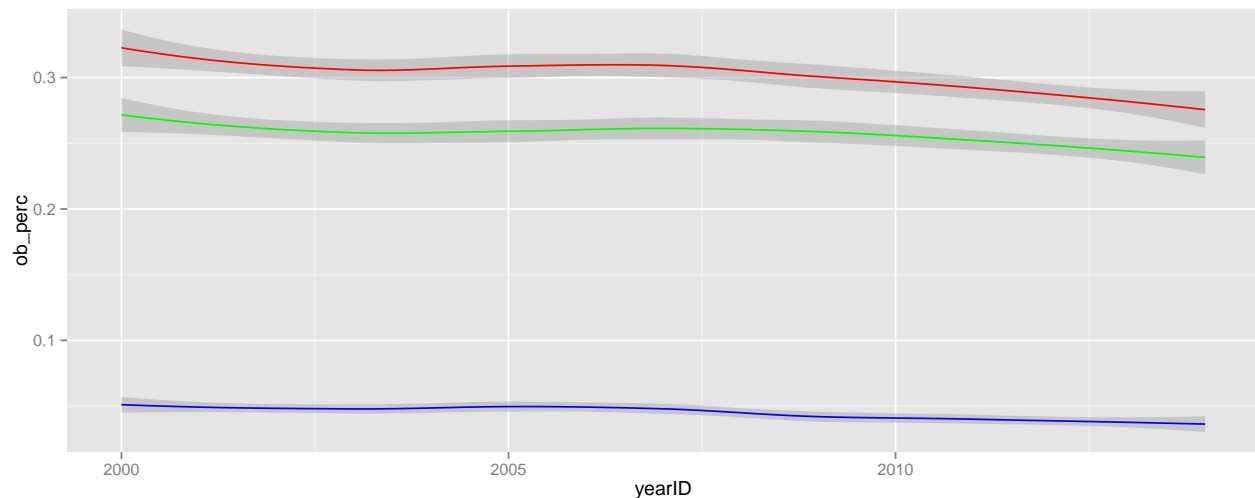
Mean on-base percentage + slugging

```
teams_bat %>%  
  group_by(yearID) %>%  
  select(ob_perc,slug_perc) %>%  
  summarise_each(funs(mean)) %>% head()
```

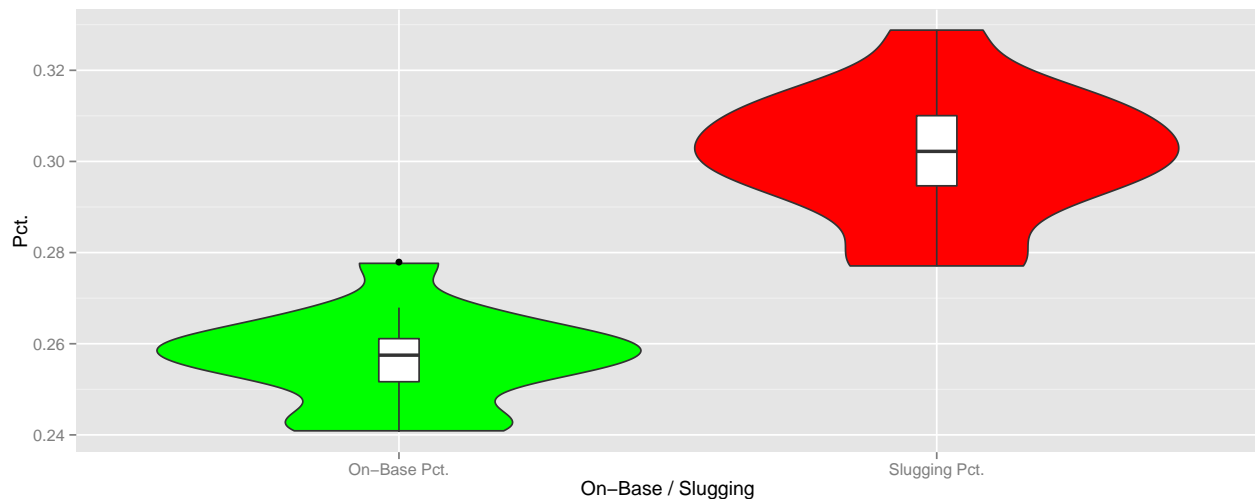
Source: local data frame [6 x 3]

	yearID	ob_perc	slug_perc
	(int)	(dbl)	(dbl)
1	2000	0.2779122	0.3288996
2	2001	0.2554038	0.3080489
3	2002	0.2574950	0.3008938
4	2003	0.2679267	0.3138558
5	2004	0.2494640	0.2970384
6	2005	0.2587080	0.3076036

```
teams_bat %>%  
  group_by(yearID) %>%  
  select(ob_perc,slug_perc) %>%  
  summarise_each(funs(mean)) %>% ggplot() +  
  geom_smooth(aes(x=yearID,y=ob_perc),color="green") +  
  geom_smooth(aes(x=yearID,y=slug_perc),color="red") +  
  geom_smooth(aes(x=yearID,y=(slug_perc - ob_perc)),color="blue")
```



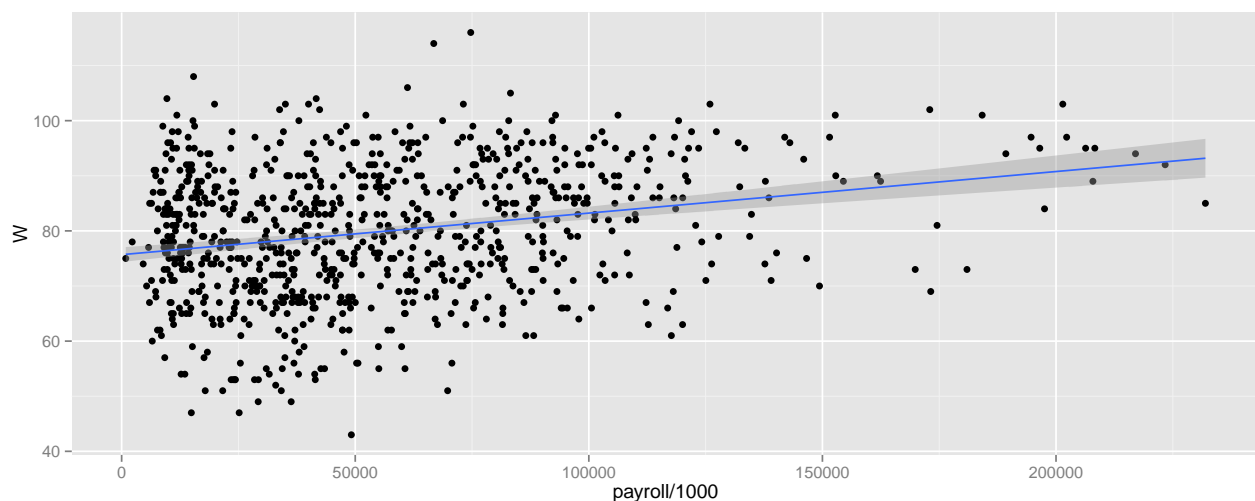
```
teams_bat %>%  
  group_by(yearID) %>%  
  select(ob_perc,slug_perc) %>%  
  summarise_each(funs(mean)) %>% ggplot() +  
  geom_violin(aes(x=factor("On-Base Pct."),y=ob_perc),fill="green") +  
  geom_violin(aes(x=factor("Slugging Pct."),y=slug_perc),fill="red") +  
  geom_boxplot(aes(x=factor("On-Base Pct."),y=ob_perc),width=0.1) +  
  geom_boxplot(aes(x=factor("Slugging Pct."),y=slug_perc),width=0.1) +  
  ylab("Pct.") +  
  xlab("On-Base / Slugging")
```



Bivariate EDA (+ even more wrangling)

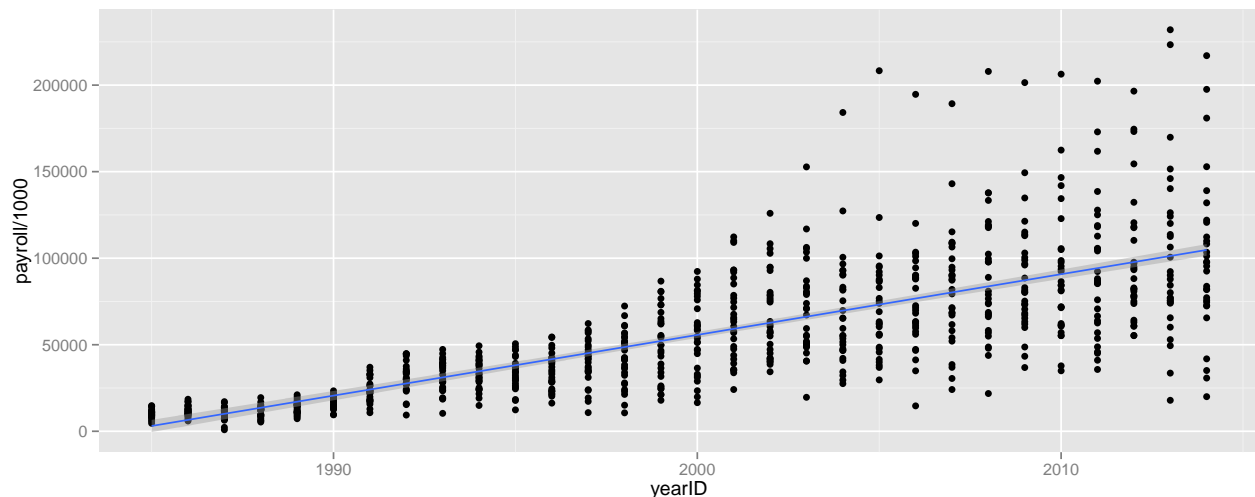
Use ggplot2 to create a scatterplot showing payroll (x-axis) and wins (y-axis) across all time periods and teams.

```
teams_pay %>%
  select payroll, W %>%
  ggplot() +
  geom_point(aes(x=payroll/1000, y=W)) +
  geom_smooth(aes(x=payroll/1000, y=W), method="lm")
```



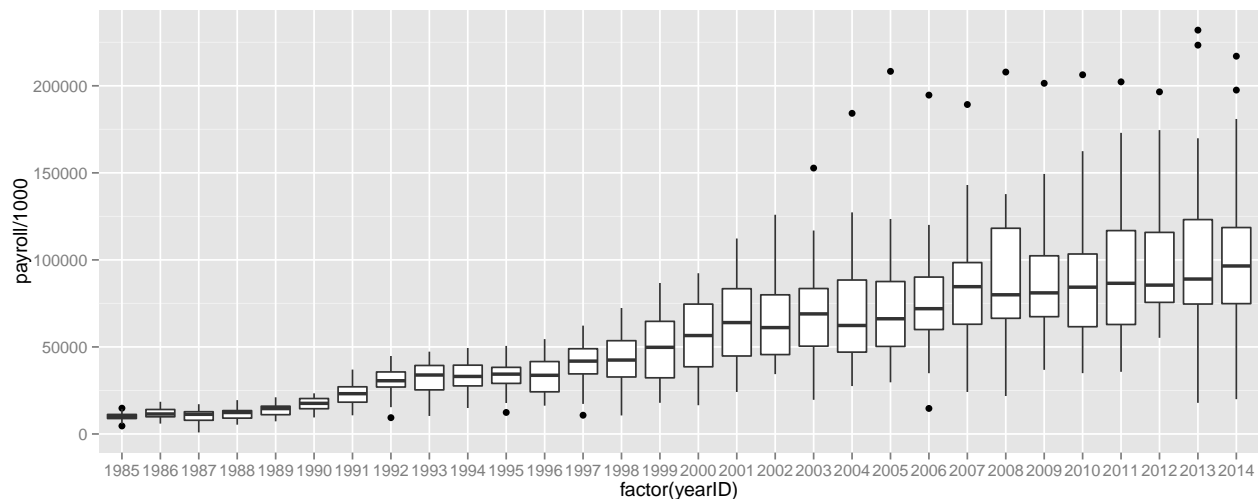
One variable we are not accounting for in this scatterplot is year. It is possible that payrolls increase from season to season. Check this out using the same ggplot code you just used above, but make this plot with year on the x-axis and payroll/1000 on the y-axis.

```
teams_pay %>%
  select(yearID, payroll) %>%
  ggplot() +
  geom_point(aes(x=yearID, y=payroll/1000)) +
  geom_smooth(aes(x=yearID, y=payroll/1000), method="lm")
```



A scatterplot may not be the best way to look at this pattern, since year is a discrete variable. So also try making boxplots stratified by yearID.

```
teams_pay %>%
  select(yearID, payroll) %>%
  ggplot() +
  geom_boxplot(aes(x=factor(yearID), y=payroll/1000))
```



Create new variables for the average payroll and the standard deviation of payrolls each year across teams and add them to your dataframe.

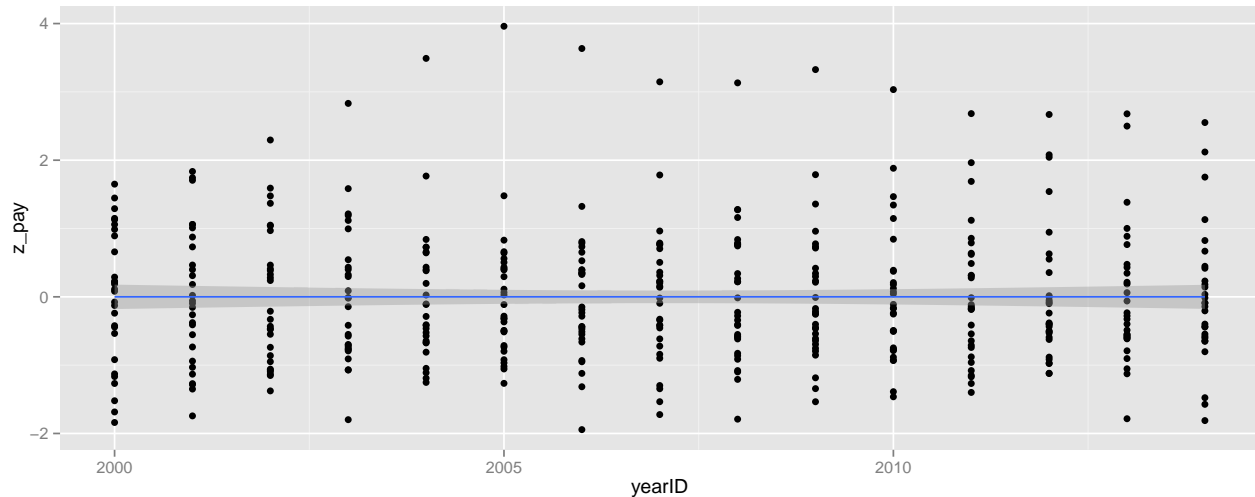
```
teams_bat <- teams_bat %>% group_by(yearID) %>%
  mutate(avg_pay = mean(payroll),
         std_pay = sd(payroll))
```

Add another variable to your dataset that is the z-score for each team for each year.

```
teams_bat <- teams_bat %>% group_by(yearID) %>%
  mutate(z_pay = (payroll - avg_pay) / std_pay)
```

Make a scatterplot in ggplot with year on the x-axis and payroll z-scores on the y-axis and two geoms: `geom_point()` and `geom_smooth(method = "lm")`.

```
teams_bat %>%
  select(yearID,z_pay) %>%
  ggplot() +
  geom_point(aes(x=yearID,y=z_pay)) +
  geom_smooth(aes(x=yearID,y=z_pay),method="lm")
```



What do you see?

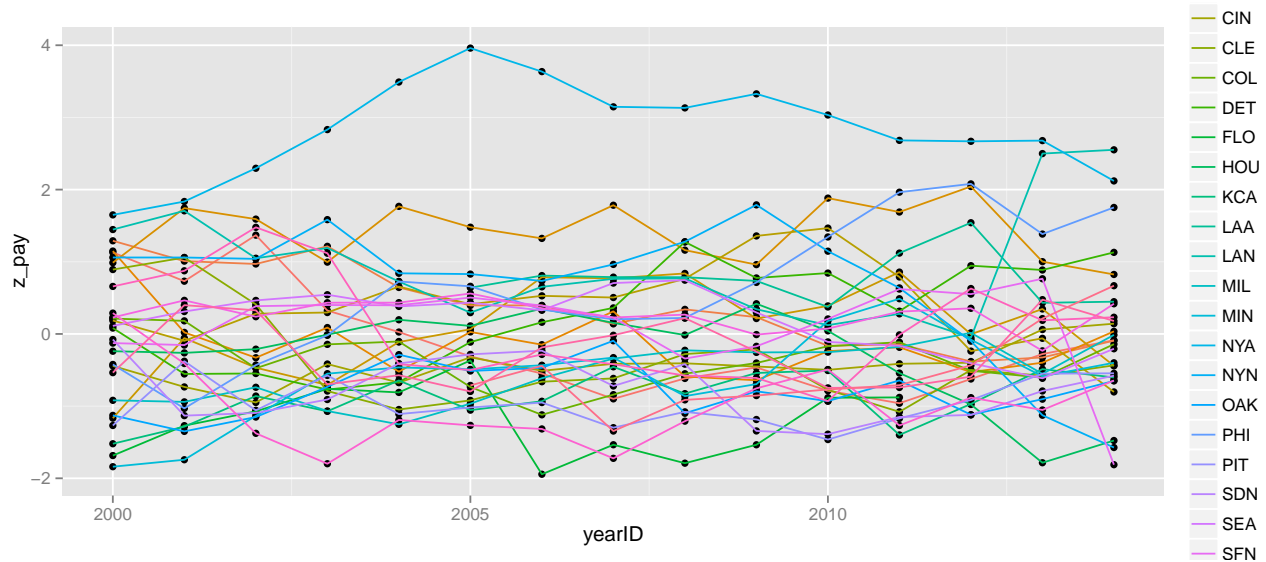
I see a scatter plot centered about $y = 0$ with a `z_pay` spread that looks similar across year. There don't appear to be obvious outliers in this data except perhaps for the Yankees.

How is this plot different from the previous one with payroll/1000 on the y-axis (from #12)?

The other plot showed payroll/1000 increasing with year. This plot shows `z_pay` steady across year (which, of course, makes perfect sense as it's relative to each year's mean payroll)

Make a new scatterplot (minus `geom_smooth`) in ggplot with year on the x-axis and payroll z-scores on the y-axis. This time, add an additional aesthetic to colour the points in the scatterplot with a different color for each `teamID`, and an additional geom called `geom_line()`.

```
teams_bat %>%
  select(yearID,z_pay,teamID) %>%
  ggplot() +
  geom_point(aes(x=yearID,y=z_pay)) +
  geom_line(aes(x=yearID,y=z_pay,color=factor(teamID)))
```



What do you see?

I see points indicating team payroll by year connected by colored lines indicating teamID.

What is not surprising here?

Teams tend to move only slightly (relative to each other) each year. Teams at the top tend to stay toward the top, teams on the bottom tend to stay toward the bottom.

Use dplyr to create a new dataset ... that includes two new variables: average payroll z-score and average number of wins. Both averages should be calculated for each team across all seasons.

```
teams_anl <- teams_bat %>%
  group_by(teamID) %>%
  summarise(avg_z_pay = mean(z_pay, na.rm=TRUE),
            avg_w_cnt = mean(W, na.rm=TRUE))
```

What will be the mean and standard deviation of this new variable across the 30 teams?

mean: This will be the mean distance of the annual payroll from the annual mean payroll across teams.

sd: This will be the spread of the distance of the annual payroll from the annual mean payroll across teams.

That is, is your new average payroll z-score also a z-score?

No.

Are you surprised?

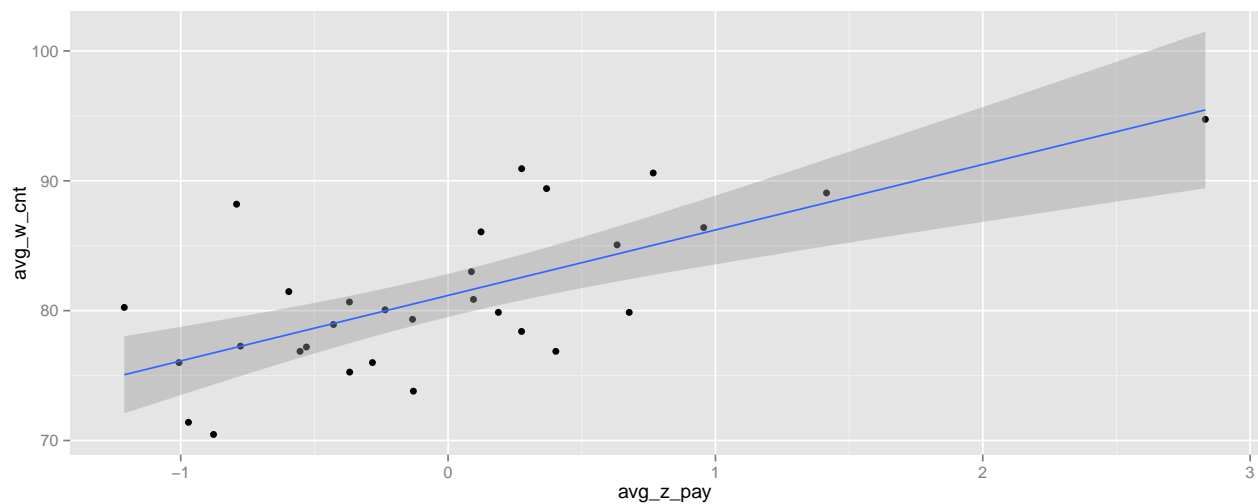
Yes!

Why or why not?

I thought about a z-score has having a unit so it made sense to me that an average of z-scores keeps its z-scoreness.

Now create a scatterplot to see the association between average payroll z-scores (x-axis) and average number of wins (y-axis).

```
teams_anl %>% ggplot() +  
  geom_point(aes(x=avg_z_pay,y=avg_w_cnt)) +  
  geom_smooth(aes(x=avg_z_pay,y=avg_w_cnt),method="lm")
```



```
teams_anl %>% lm(avg_w_cnt ~ avg_z_pay,.) %>% summary()
```

Call:

```
lm(formula = avg_w_cnt ~ avg_z_pay, data = .)
```

Residuals:

Min	1Q	Median	3Q	Max
-6.7168	-3.3719	-0.0858	1.3803	11.0187

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	81.1713	0.8122	99.937	< 2e-16 ***
avg_z_pay	5.0438	0.9974	5.057	2.37e-05 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.449 on 28 degrees of freedom

Multiple R-squared: 0.4773, Adjusted R-squared: 0.4587

F-statistic: 25.57 on 1 and 28 DF, p-value: 2.375e-05

```
cor(teams_anl$avg_z_pay,teams_anl$avg_w_cnt)
```

```
[1] 0.6908895
```


Use both the plot and the correlation statistics to evaluate (in words) the form (does the relationship look linear?) and strength of the association between these two variables.

The form looks as if it can be approximated by a linear model. The correlation is high ($\text{cor} = 0.6908895$, $\text{R-squared} = 0.4773283$) but significant ($\text{p-value} = 2.375\text{e-}05$).

Would you be comfortable using a linear model to predict the mean number of wins in a given season given their average relative payroll for that season?

Given such a low R-squared (not close to 1), I'd be hesitant to predict the exact number of wins, though with such a low p-value, I might be comfortable making more general predictions, like predicting the mean number of wins is within some range etc.

Regression model

Build a simple linear regression model predicting mean wins from mean payroll z-scores across seasons.

```
teams_anl_lm <- teams_anl %>% lm(avg_w_cnt ~ avg_z_pay,.)
teams_anl_lm %>% anova()
```

Analysis of Variance Table

```
Response: avg_w_cnt
      Df Sum Sq Mean Sq F value    Pr(>F)
avg_z_pay  1  506.05   506.05   25.571 2.375e-05 ***
Residuals 28  554.13    19.79
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
teams_anl_lm %>% summary()
```

Call:

```
lm(formula = avg_w_cnt ~ avg_z_pay, data = .)
```

Residuals:

Min	1Q	Median	3Q	Max
-6.7168	-3.3719	-0.0858	1.3803	11.0187

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	81.1713	0.8122	99.937	< 2e-16 ***
avg_z_pay	5.0438	0.9974	5.057	2.37e-05 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.449 on 28 degrees of freedom

Multiple R-squared: 0.4773, Adjusted R-squared: 0.4587

F-statistic: 25.57 on 1 and 28 DF, p-value: 2.375e-05

```
teams_anl_lm %>% glance()
```

```
   r.squared adj.r.squared   sigma statistic   p.value df   logLik
1 0.4773283    0.4586614 4.448623  25.57091 2.374621e-05  2 -86.3111
   AIC      BIC deviance df.residual
1 178.6222 182.8258 554.1268         28
```

What are the total, model, and residual sums of squares for this simple linear regression?

model: 506.05
residual: 554.13

What percent of the variation in mean wins is “explained” by variation in mean payroll z-scores?

47.73283% (R-squared)

Write up a summary of your findings.

The results indicate that payroll predicted wins ($b_1 = 5.0438$), with 47.73283% of the variance in wins accounted for by payroll levels. Each standard deviation payroll was from the annual mean was associated with an increase of 5.0434 wins. The OLS regression equation for predicting wins is of the form

$$\text{wins}_i = 81.1713 + 5.0438 \text{standard deviations from mean payroll}_i + \epsilon_i$$

What is the average of all \hat{y}_i values (in any simple linear regression model) equal to?

The mean response (intercept).

What is the variance of the residuals in your regression model?

19.107

The standard error?

4.449

Compare the variance of the residuals to sample variance of mean wins overall, and to your model R².

variance of residuals: 19.79 variance of mean wins overall: 36.55798 R-squared: 0.4773 Adj.
R-squared = 0.4587

$$1 - \frac{19.79}{36.55798} \approx 0.4587$$

How are these three statistics related (in any simple linear regression model)?

$$1 - \frac{\text{variance of residuals}}{\text{variance of mean wins overall}} = \text{Adj. R-squared}$$

Obviously, the book and movie about the Oakland A's suggests that this team may be an outlier in terms of the predicting wins from payroll. Look specifically at this team:

What is the observed number of mean wins?

```
teams_anl %>% filter(teamID == "OAK")
```

Source: local data frame [1 x 3]

	teamID (chr)	avg_z_pay (dbl)	avg_w_cnt (dbl)
1	OAK	-0.7910688	88.2

88.2

What is the predicted?

$$81.1713 + 5.0438 \times -0.7910688 \approx 77.18$$

What is the residual?

$$88.2 - 77.18 = 11.02$$

How many standard deviations above/below the residual mean is the Oakland A's residual value?

```
augment(teams_anl_lm) %>% filter(avg_w_cnt == 88.2)
```

	avg_w_cnt	avg_z_pay	.fitted	.se.fit	.resid	.hat	.sigma
1	88.2	-0.7910688	77.18132	1.128597	11.01868	0.06436152	3.964489
			.cooksd	.std.resid			
1	0.2255216	2.560649					

2.56 above

Are there any other teams with a residual value as extreme or more extreme than the Oakland A's?

```
augment(teams_anl_lm) %>% arrange(.std.resid) %>% tail()
```

	avg_w_cnt	avg_z_pay	.fitted	.se.fit	.resid	.hat	.sigma
25	86.06667	0.1232494	81.79296	0.8222835	4.273711	0.03416583	4.452282
26	80.25000	-1.2111056	75.06274	1.4511598	5.187262	0.10640924	4.405447
27	90.60000	0.7672145	85.04099	1.1196456	5.559013	0.06334466	4.393321
28	89.40000	0.3683245	83.02907	0.8936759	6.370934	0.04035608	4.353930
29	90.93333	0.2751527	82.55913	0.8590704	8.374207	0.03729120	4.222002
30	88.20000	-0.7910688	77.18132	1.1285966	11.018681	0.06436152	3.964489
			.cooksd	.std.resid			
25	0.01690114	0.9775259					

```

26 0.09059354 1.2335117
27 0.05637217 1.2911665
28 0.04493801 1.4619162
29 0.07128904 1.9185392
30 0.22552164 2.5606486

```

No. The next highest is SLN (St. Louis Cardinals) with 1.92

Create a bootstrap distribution for the correlation and the regression coefficients. Copy and paste the following code into your file, and annotate each line with a # to (briefly) explain what each line of code is doing.

```

orig_cor <- 0.6908895
orig_slp <- 5.0438
gt_cor_cnt <- 0
gt_slp_cnt <- 0
N <- 10^4 # storing 10000 as N
cor.boot <- numeric(N) # store vector of size 10000 as cor.boot
int.boot <- numeric(N) # store vector of size 10000 as int.boot
slope.boot <- numeric(N) # store vector of size 10000 as slope.boot
n <- 30 # number of observations here
for (i in 1:N){ # loop 10000 times, storing loop iteration as i
  index <- sample(n, replace = TRUE) # store a vector of size n
                                     # with values ranging from 1
                                     # to n as index

  team.boot <- teams_anl[index, ] # resampled data
  cor.boot[i] <- cor(team.boot$avg_z_pay, team.boot$avg_w_cnt)
  # what is x and y? The input & response variables, avg_z_pay and avg_w_cnt
  if(cor.boot[i] > orig_cor){
    gt_cor_cnt <- gt_cor_cnt + 1
  }
  # recalculate linear model estimates
  team.boot.lm <- lm(avg_w_cnt ~ avg_z_pay, data = team.boot)
  # what is x and y?
  int.boot[i] <- coef(team.boot.lm)[1] # new intercept
  slope.boot[i] <- coef(team.boot.lm)[2] # new slope
  if(slope.boot[i] > orig_slp){
    gt_slp_cnt <- gt_slp_cnt + 1
  }
}

mean(cor.boot) #mean correlation of bootstrapped data

```

```
[1] 0.6772078
```

```
sd(cor.boot) #standard deviation of correlation of bootstrapped data
```

```
[1] 0.1192985
```

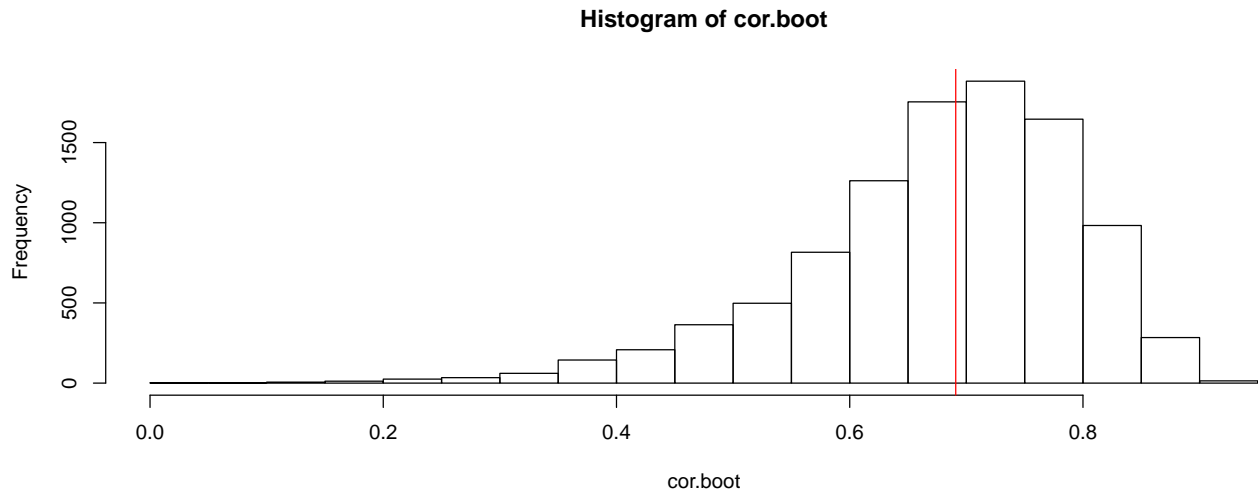
```
quantile(cor.boot, c(.025, .975)) #95% CI of correlation of bootstrapped data
```

```

      2.5%      97.5%
0.3880698 0.8537613

```

```
hist(cor.boot)
#create histogram of correlation of bootstrapped data
observed <- cor(teams_anl$avg_z_pay, teams_anl$avg_w_cnt)
# what is x and y? The input & response variables avg_z_pay and avg_w_cnt
abline(v = observed, col = "red") # add line at original sample correlation
```



```
# do the same as above for slope.boot (don't worry about int.boot)
mean(slope.boot) #mean slope of bootstrapped data
```

```
[1] 5.099125
```

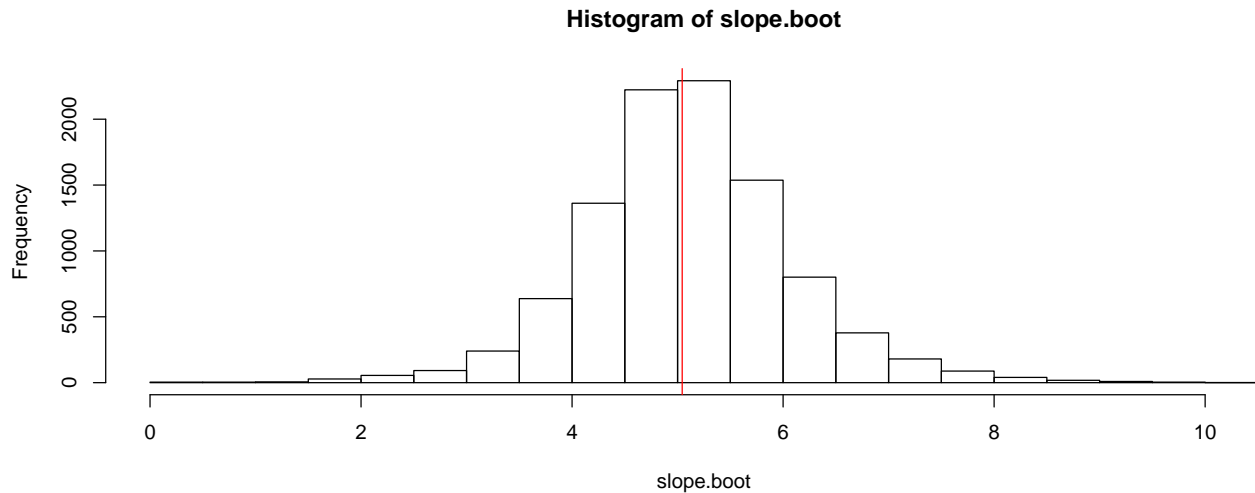
```
sd(slope.boot) #standard deviation of slope of bootstrapped data
```

```
[1] 0.9864231
```

```
quantile(slope.boot, c(.025, .975)) #95% CI of slope of bootstrapped data
```

```
      2.5%      97.5%
3.149532 7.241109
```

```
hist(slope.boot) #create histogram of slope of bootstrapped data
observed <- summary(teams_anl_lm)$coefficients[2]
# what is x and y? The input & response variables avg_z_pay and avg_w_cnt
abline(v = observed, col = "red") # add line at original sample slope
```



```
gt_cor_cnt
```

```
[1] 5171
```

```
gt_slp_cnt
```

```
[1] 5123
```

Figure out how many bootstrap samples had a higher correlation than the one you observed as your original sample correlation.

5198 (51.98%)

How many bootstrap samples had a higher slope coefficient than the one you observed.

5065 (50.65%)

Use `dplyr::mutate()` with `ifelse()` to create a categorical variable that splits our `yearID` variable into two time intervals: 2000 - 2006 and 2007 - 2014. Then look at your work for question 14 and update to re-calculate average wins and average payroll z-scores separately for each team and time interval (hint: that means two variables in a `dplyr::group_by()` statement).

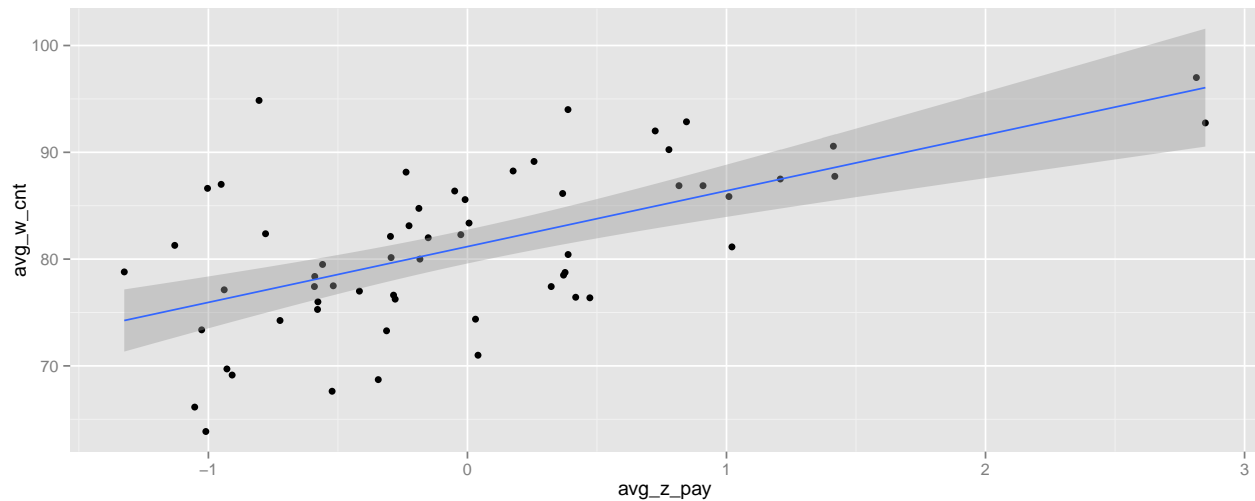
```
teams_bat_year_split <- teams_bat %>% # rename these dataframes as appropriate
  mutate(recent = ifelse(yearID < 2007, 0, 1)) # rename variables as appropriate
table(teams_bat_year_split$recent, teams_bat_year_split$yearID) # trust but verify
```

	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013
0	28	28	28	28	28	30	30	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	30	30	30	30	30	29	29

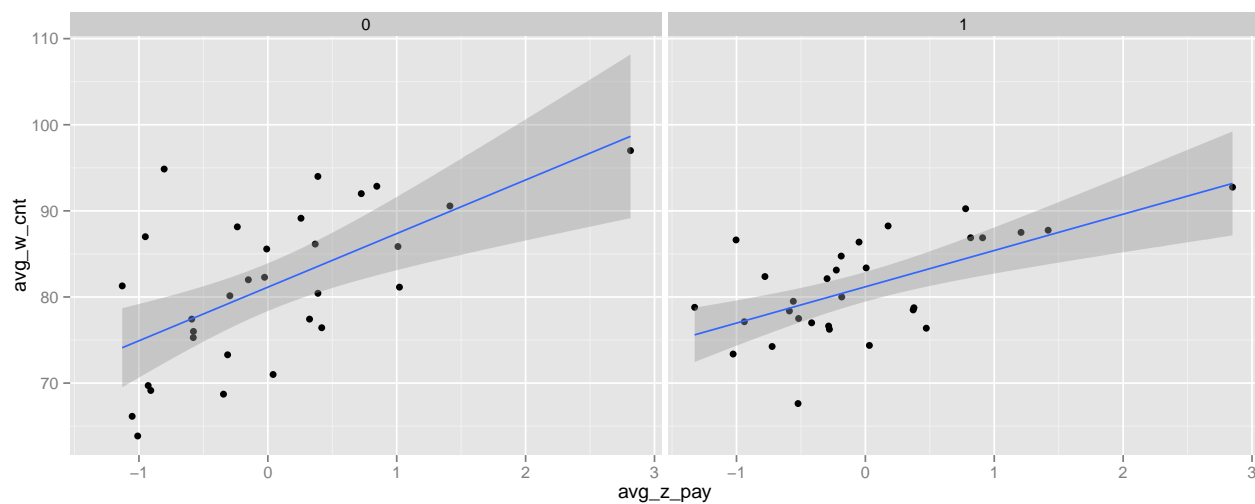
	2014
0	0
1	29

```
teams_bat_year_split_anl <- teams_bat_year_split %>%
  group_by(teamID,recent) %>%
  summarise(avg_z_pay = mean(z_pay,na.rm=TRUE),
    avg_w_cnt = mean(W,na.rm=TRUE))
```

```
teams_bat_year_split_anl %>% group_by(teamID) %>%
  ggplot() +
  geom_point(aes(x=avg_z_pay,y=avg_w_cnt)) +
  geom_smooth(aes(x=avg_z_pay,y=avg_w_cnt),method="lm")
```



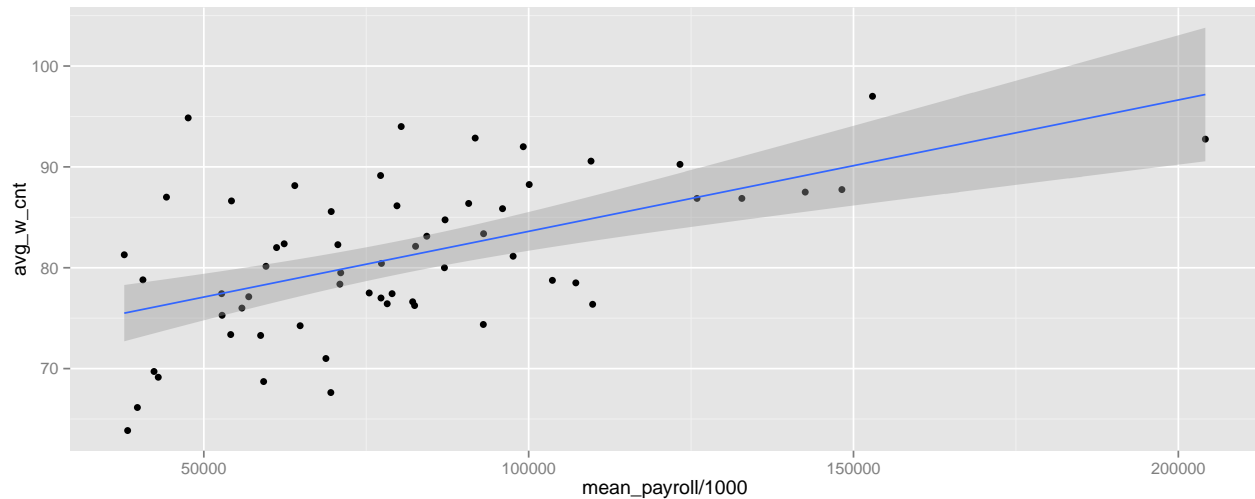
```
teams_bat_year_split_anl %>% group_by(teamID) %>%
  ggplot() +
  facet_wrap(~ recent) +
  geom_point(aes(x=avg_z_pay,y=avg_w_cnt)) +
  geom_smooth(aes(x=avg_z_pay,y=avg_w_cnt),method="lm")
```



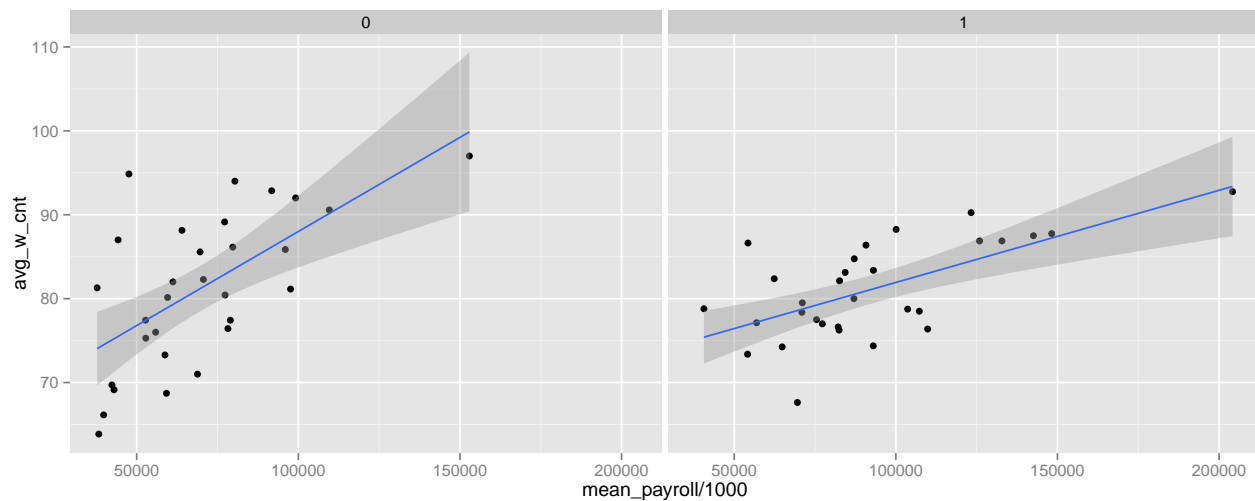
Using ggplot2, create one plot, with side-by-side scatterplots for each time interval, showing mean payroll (x-axis) and mean number of wins (y-axis) across all teams.

```
teams_bat_year_split_anl_w_mean_payroll <- teams_bat_year_split %>%
  group_by(teamID,recent) %>%
  summarise(mean_payroll = mean(payroll,na.rm=TRUE),
    avg_w_cnt = mean(W,na.rm=TRUE))
```

```
teams_bat_year_split_anl_w_mean_payroll %>% group_by(teamID) %>%
  ggplot() +
  geom_point(aes(x=mean_payroll/1000,y=avg_w_cnt)) +
  geom_smooth(aes(x=mean_payroll/1000,y=avg_w_cnt),method="lm")
```



```
teams_bat_year_split_anl_w_mean_payroll %>% group_by(teamID,recent) %>%
  ggplot() +
  facet_wrap(~ recent) +
  geom_point(aes(x=mean_payroll/1000,y=avg_w_cnt)) +
  geom_smooth(aes(x=mean_payroll/1000,y=avg_w_cnt),method="lm")
```



Comment on differences you see between these two plots, and compare to your previous scatterplot across all seasons.

It appears the slope of the regression line was steeper before 2007, indicating wins were cheaper.

The scatterplot across all seasons hides the fact that these two quite models exist and simply draws a regression line over all the datapoints.

Now, run two linear regression analyses (as shown in class), one for each time interval, using `dplyr::group_by()` `%>% do()` and `broom::tidy()/glance()/augment()`.

```
models <- teams_bat_year_split_anl %>%
  group_by(recent) %>%
  do(mod = lm(avg_w_cnt ~ avg_z_pay, data = .))

models %>% tidy(mod) #coefs
```

Source: local data frame [4 x 6]
Groups: recent [2]

	recent (dbl)	term (chr)	estimate (dbl)	std.error (dbl)	statistic (dbl)	p.value (dbl)
1	0	(Intercept)	81.140032	1.3542026	59.917201	4.145641e-31
2	0	avg_z_pay	6.225057	1.5798253	3.940345	4.929825e-04
3	1	(Intercept)	81.183945	0.8403026	96.612750	6.856896e-37
4	1	avg_z_pay	4.213821	0.9859629	4.273813	2.008604e-04

```
models %>% augment(mod) %>%
  group_by(recent) %>%
  summarize(tot_ss = sum((avg_w_cnt - mean(avg_w_cnt))^2),
            res_ss = sum((avg_w_cnt - .fitted)^2))
```

Source: local data frame [2 x 3]

	recent (dbl)	tot_ss (dbl)	res_ss (dbl)
1	0	2394.6014	1540.4205
2	1	979.6834	592.9072

```
models %>% glance(mod)
```

Source: local data frame [2 x 12]
Groups: recent [2]

	recent (dbl)	r.squared (dbl)	adj.r.squared (dbl)	sigma (dbl)	statistic (dbl)	p.value (dbl)	df (int)
1	0	0.3567111	0.3337365	7.417211	15.52632	0.0004929825	2
2	1	0.3947971	0.3731827	4.601658	18.26548	0.0002008604	2

Variables not shown: logLik (dbl), AIC (dbl), BIC (dbl), deviance (dbl), df.residual (int)

Compare the coefficient estimates to each other, and to your original model.

The intercepts of both models are similar (81.14 and 81.18) but the slopes are not (6.22 and 4.21). The coefficients of the original model are between these two models' coefficients (original intercept = 81.17 and original slope = 5.04).

Take the Oakland A's team as a specific case:

Which of your three model/time interval regression models (model 1: across all seasons; model 2: 2000 - 2006; model 3: 2007 - 2014) was better at predicting mean wins for them specifically?

```
#Oakland A's actual data
```

```
avg_w_cnt = 88.2
```

```
avg_z_pay = -0.7910688
```

```
#model 1
```

```
(81.17 + 5.04 * avg_z_pay) - avg_w_cnt
```

```
[1] -11.01699
```

```
#model 2
```

```
(81.14 + 6.22 * avg_z_pay) - avg_w_cnt
```

```
[1] -11.98045
```

```
#model 3
```

```
(81.18 + 4.21 * avg_z_pay) - avg_w_cnt
```

```
[1] -10.3504
```

model 3

Which model overall accounted for the most variability in mean wins overall across all teams?

model 1. It had an R-squared of 0.4773 while model 2 and 3 had R-squared values of 0.3567 and 0.3947, respectively.

How is the R2 estimate related to the plain old correlation between average wins and average payroll z-scores for each time interval?

$$0.4773 = 0.690869^2$$

$$0.3567 = 0.5972437^2$$

$$0.3947 = 0.6282515^2$$

And in general in any simple linear regression model?

R-squared is the square of the correlation.

Midterm: Exercises

Part 1: Probability

The following table shows the cumulative distribution function of a discrete random variable. Find the probability mass function.

```
k = c(0,1,2,3,4,5)
F.k = c(0,.1,.3,.7,.8,1.0)
df = data.frame(k,F.k)
kable(df)
```

k	F.k
0	0.0
1	0.1
2	0.3
3	0.7
4	0.8
5	1.0

$$\text{cdf: } F(x) = \sum_{t \leq x} f(t)$$

$$\text{pmf: } f(t) = F(x) - F(x-1) = \sum_{t \leq x} f(t) - \sum_{t \leq x-1} f(t)$$

$$\begin{aligned}
 f(0) &= F(0) = 0 \\
 f(1) &= F(1) - F(0) = .1 - 0 = .1 \\
 f(2) &= F(2) - F(1) = .3 - .1 = .2 \\
 f(3) &= F(3) - F(2) = .7 - .3 = .4 \\
 f(4) &= F(4) - F(3) = .8 - .7 = .1 \\
 f(5) &= F(5) - F(4) = 1.0 - .8 = .2
 \end{aligned}$$

```
k = c(0,1,2,3,4,5)
F.k = c(0,.1,.3,.7,.8,1.0)
f.k = c(0,.1,.2,.4,.1,.2)
df = data.frame(k,F.k,f.k)
kable(df)
```

k	F.k	f.k
0	0.0	0.0
1	0.1	0.1
2	0.3	0.2
3	0.7	0.4
4	0.8	0.1
5	1.0	0.2

The probability density function of a random variable X is given by:

$$f(x) = \begin{cases} cx, & 0 < x < 4 \\ 0, & \text{otherwise} \end{cases}$$

a) find c

$$\int_{-\infty}^{\infty} f_x(x) dx = 1$$

$$\begin{aligned} 1 &= \int_0^4 cx \, dx \\ &= c \times \left[\frac{x^2}{2} \right]_0^4 \\ &= c \times \left[\frac{4^2}{2} - 0 \right] \\ &= c \times \left[\frac{16}{2} \right] \\ &= 8c \end{aligned}$$

$$c = \frac{1}{8}$$

b) find the cumulative distribution function $F(x)$.

$$F_x(x) = \int_{-\infty}^x f_x(y) \, dy = P(X \leq x)$$

$$\begin{aligned} F_x(x) &= \int_0^x \frac{1}{8}(y) \, dy \text{ when } 0 \leq x \leq 4 \\ &= \frac{1}{8} \left[\frac{y^2}{2} \right]_0^x \\ &= \frac{1}{8} \left[\frac{x^2}{2} - 0 \right] \\ &= \frac{x^2}{16} \end{aligned}$$

$$F_x(x) = \begin{cases} 0, & x < 0 \\ \frac{x^2}{16}, & 0 \leq x \leq 4 \\ 1, & x > 4 \end{cases}$$

c) Compute $P(1 < X < 3)$

$$\begin{aligned}
P(1 < X < 3) &= F_x(3) - F_x(1) \\
&= \frac{3^2}{16} - \frac{1^2}{16} \\
&= \frac{9}{16} - \frac{1}{16} \\
&= \frac{8}{16} \\
&= \frac{1}{2} \\
&= 0.5
\end{aligned}$$

The random variable X has a cumulative distribution function (cdf):

$$F(x) = \begin{cases} 0, & x \leq 0 \\ \frac{x^3}{2+x^2}, & x > 0 \end{cases}$$

Find the probability density function (pdf) of X.

$$\begin{aligned}
pdf(x) &= f_x(x) = F'(x) \\
&= \frac{x^3}{2+x^2}' \text{ when } x > 0 \\
&= \frac{(2+x^2) \times 3x^2 - x^3 \times 2x}{(2+x^2)^2} \text{ (quotient rule)} \\
&= \frac{6x^2 + 3x^4 - 2x^4}{(2+x^2)^2} \\
&= \frac{6x^2 + x^4}{(2+x^2)^2} \\
&= \frac{x^2(6+x^2)}{(2+x^2)^2}
\end{aligned}$$

$$f_x(x) = \begin{cases} 0, & x \leq 0 \\ \frac{x^2(6+x^2)}{(2+x^2)^2}, & x > 0 \end{cases}$$

The joint probability of the continuous random variable (X, Y) is given by:

$$f(x, y) = \begin{cases} \frac{1}{28}(4x + 2y + 1), & 0 \leq x < 2, 0 \leq y < 2 \\ 0, & \text{otherwise} \end{cases}$$

Find E(XY)

$$\begin{aligned}
E[XY] &= \int_0^2 \int_0^2 xy \left[\frac{1}{28}(4x + 2y + 1) \right] dx dy \\
&= \frac{1}{28} \int_0^2 \int_0^2 4x^2y + 2xy^2 + xy dx dy
\end{aligned}$$

$$\begin{aligned}
(\text{expr. 1}) &= \frac{1}{28} \int_0^2 4x^2y + 2xy^2 + xy \, dx \\
&= \frac{1}{28} \left[4\frac{x^3}{3}y + 2\frac{x^2}{2}y^2 + \frac{x^2}{2}y \right]_{x=0}^{x=2} \\
&= \frac{1}{28} \left[4\frac{8}{3}y + 2 \times 2y^2 + 2y \right] \\
&= \frac{1}{28} \left[\frac{32y}{3} + 4y^2 + 2y \right] \\
&= \frac{1}{28} \left[\frac{38y}{3} + 4y^2 \right] \\
&= \frac{38y}{84} + \frac{4y^2}{28} \\
&= \frac{19y}{42} + \frac{y^2}{7}
\end{aligned}$$

$$\begin{aligned}
E[XY] &= \int_0^2 (\text{expr. 1}) \, dy \\
&= \int_0^2 \frac{19y}{42} + \frac{y^2}{7} \, dy \\
&= \left[\frac{19}{42} \times \frac{y^2}{2} + \frac{1}{7} \times \frac{y^3}{3} \right]_0^2 \\
&= \left[\frac{19}{42} \times 2 + \frac{1}{7} \times \frac{8}{3} \right] - 0 \\
&\approx 1.286
\end{aligned}$$

Find Cov(X, Y)

Marginal Probabilities:

$$\begin{aligned}
f_x(x) &= \int_{-\infty}^{\infty} f(x, y) \, dy \\
f_y(y) &= \int_{-\infty}^{\infty} f(x, y) \, dx
\end{aligned}$$

$$\begin{aligned}
f_x(x) &= \int_0^2 \frac{1}{28} (4x + 2y + 1) \, dy \\
&= \frac{1}{28} \left[4xy + 2\frac{y^2}{2} + y \right]_0^2 \\
&= \frac{1}{28} [8x + 4 + 2] \\
&= \frac{8x}{28} + \frac{6}{28} \\
&= \frac{2x}{7} + \frac{3}{14}
\end{aligned}$$

$$\begin{aligned}
 f_y(y) &= \int_0^2 \frac{1}{28}(4x + 2y + 1) dx \\
 &= \frac{1}{28} \left[4 \frac{x^2}{2} + 2yx + x \right]_0^2 \\
 &= \frac{1}{28} [8 + 4y + 2] \\
 &= \frac{4y}{28} + \frac{10}{28} \\
 &= \frac{y}{7} + \frac{5}{14}
 \end{aligned}$$

$$\begin{aligned}
 E[X] &= \int_{-\infty}^{\infty} x f_x(x) dx \\
 &= \int_0^2 x \left[\frac{2x}{7} + \frac{3}{14} \right] dx \\
 &= \int_0^2 \frac{2x^2}{7} + \frac{3x}{14} dx \\
 &= \left[\frac{2x^3}{21} + \frac{3x^2}{28} \right]_0^2 \\
 &= \left[\frac{16}{21} + \frac{12}{28} \right] - 0 \\
 &\approx 1.190
 \end{aligned}$$

$$\begin{aligned}
 E[Y] &= \int_{-\infty}^{\infty} y f_y(y) dy \\
 &= \int_0^2 y \left[\frac{y}{7} + \frac{5}{14} \right] dy \\
 &= \int_0^2 \frac{y^2}{7} + \frac{5y}{14} dy \\
 &= \left[\frac{y^3}{21} + \frac{5y^2}{28} \right]_0^2 \\
 &= \left[\frac{8}{21} + \frac{20}{28} \right] - 0 \\
 &\approx 1.095
 \end{aligned}$$

$$\begin{aligned}
 \text{Cov}(X, Y) &= E[XY] - E[X]E[Y] \\
 &\approx 1.286 - 1.095 \times 1.190 \\
 &\approx 1.286 - 1.30305 \\
 &\approx -0.01705
 \end{aligned}$$

Find the correlation coefficient ρ_{XY}

$$\begin{aligned}
Var(X) &= \int_{-\infty}^{\infty} (X - \mu x)^2 f(x) dx \\
&= \int_0^2 (X - 1.190)^2 \times \left[\frac{2x}{7} + \frac{3}{14} \right] dx \\
&\quad \text{used wolfram alpha solver} \\
&\quad \approx 0.297
\end{aligned}$$

$$\begin{aligned}
Var(Y) &= \int_{-\infty}^{\infty} (Y - \mu y)^2 f(y) dy \\
&= \int_0^2 (Y - 1.095)^2 \times \left[\frac{y}{7} + \frac{5}{14} \right] dy \\
&\quad \text{used wolfram alpha solver} \\
&\quad \approx 0.324
\end{aligned}$$

$$\begin{aligned}
Cor(X, Y) &= \frac{Cov(X, Y)}{\sqrt{Var(X) \times Var(Y)}} \\
&\quad \approx \frac{-0.017}{\sqrt{0.297 \times 0.324}} \\
&\quad \approx -0.055
\end{aligned}$$

Part 2: Sampling Distributions

Examine the behavior of a lognormal random variable with parameters 0.2938933 and 1.268636.

```
set.seed(12345)
logn_1samp <- rlnorm(1e+07, 0.2938933, 1.268636)
mean(logn_1samp)
```

```
[1] 3.002705
```

```
sd(logn_1samp)
```

```
[1] 6.036241
```

Transform this variable linearly so that we have a new variable Y mean of 100 and a standard deviation of 15.

```
set.seed(12345)
logn_1samp <- 2.5 * rlnorm(1e+07, 0.2938933, 1.268636) + 92.5
mean(logn_1samp)
```

```
[1] 100.0068
```

```
sd(logn_1samp)
```

```
[1] 15.0906
```

Take 100,000 means based on samples of size 25 from the transformed lognormal distribution.


```

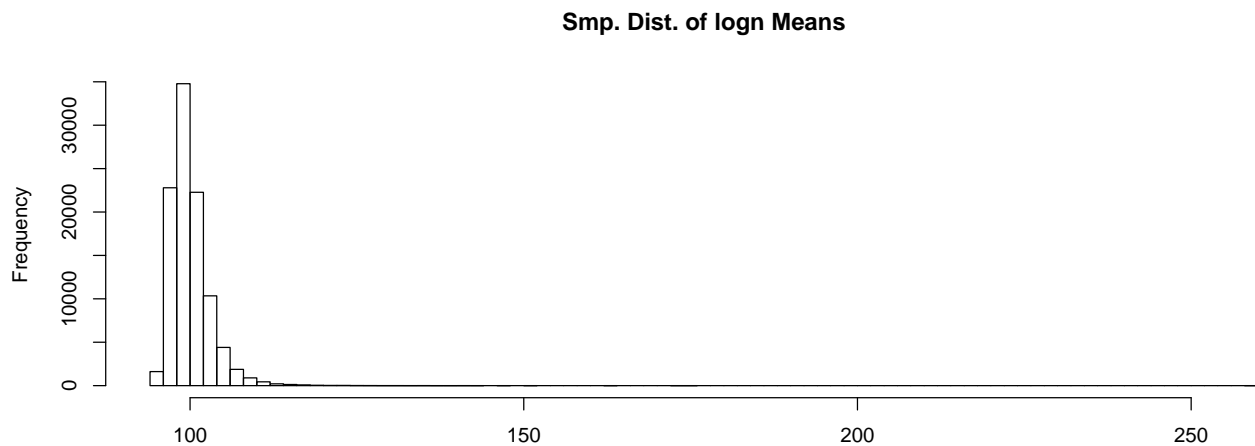
set.seed(12345)
N = 100000
logn_means <- numeric(N)
for (i in 1:N) {
  x <- 2.5 * rlnorm(25, 0.2938933, 1.268636) + 92.5
  logn_means[i] <- mean(x)
}
head(logn_means)

```

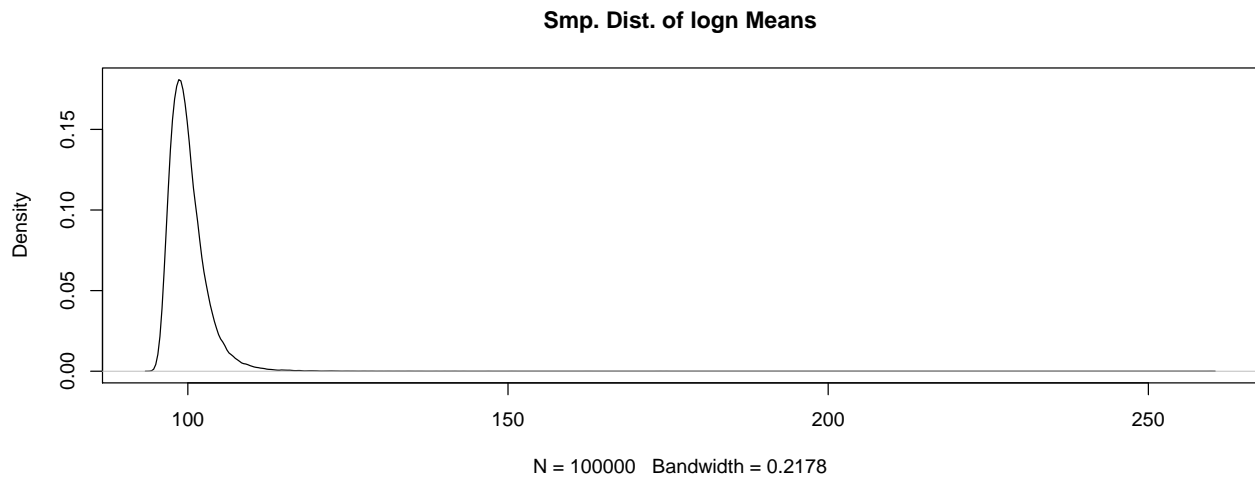
```
[1] 98.69147 104.86052 104.36481 105.31974 101.71476 101.00076
```

Examine the population, sample, and sampling distributions.

```
logn_means %>% hist(breaks=100,main="Smp. Dist. of logn Means")
```



```
plot(density(logn_means),main="Smp. Dist. of logn Means")
```



What did you expect to see?

I suppose I expected the means to be evenly distributed above and below 100.

What do you actually see?

A high peak around 100 with a long tail to the right.

What is the mean/standard deviation of this simulated sampling distribution?

```
mean(logn_means)
```

```
[1] 100.0029
```

```
sd(logn_means)
```

```
[1] 3.023481
```

mean: 100.0029 sd: 3.023481, odd. didn't we transform this to 15?

Do the same for an exponential distribution with mean and standard deviation of 1.

```
set.seed(12345)
exp_1samp <- rexp(1e+07,1)
mean(exp_1samp)
```

```
[1] 0.9997628
```

```
sd(exp_1samp)
```

```
[1] 0.9995945
```

```
set.seed(12345)
exp_1samp <- 15 * rexp(1e+07,1) + 92.5
mean(exp_1samp)
```

```
[1] 107.4964
```

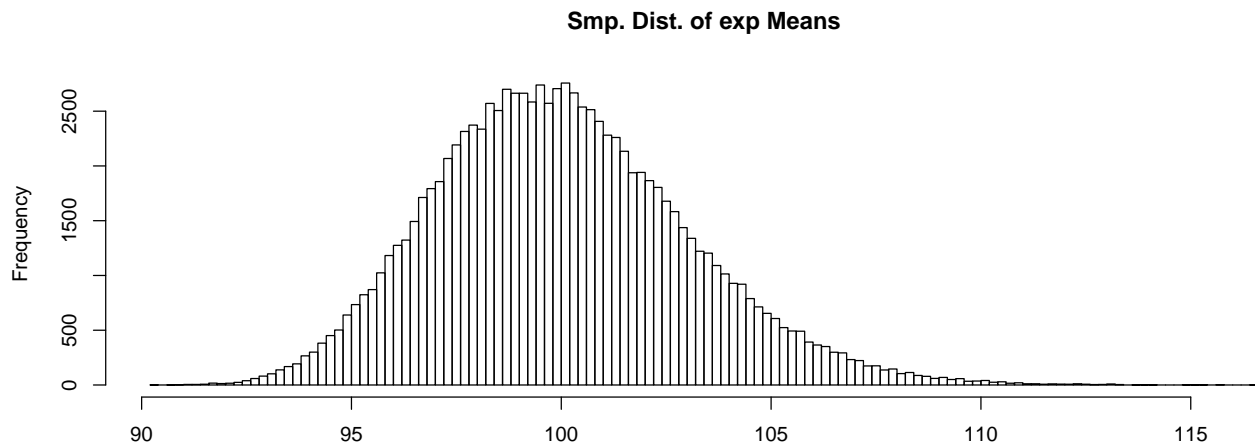
```
sd(exp_1samp)
```

```
[1] 14.99392
```

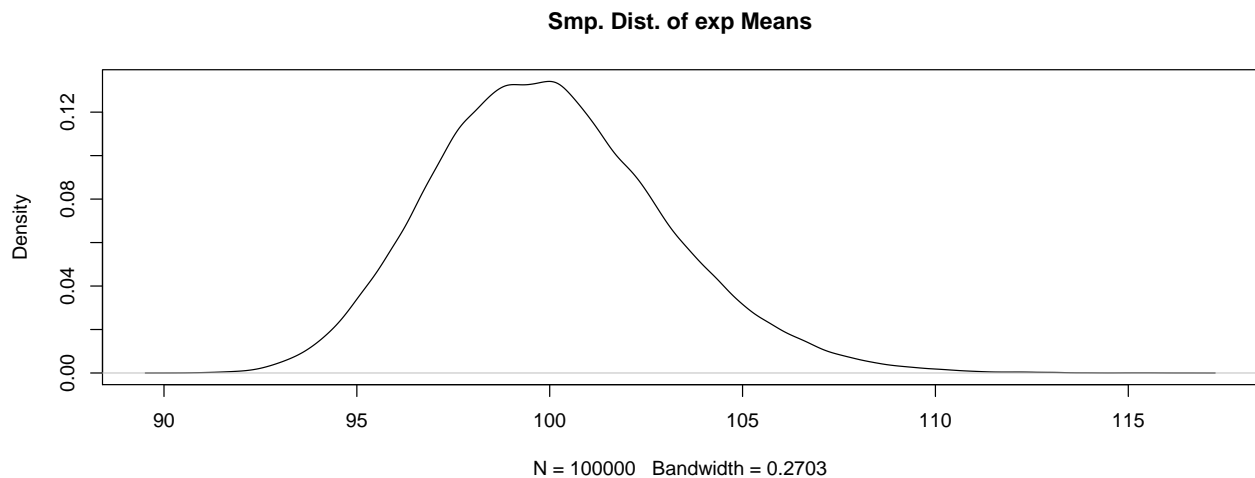
```
set.seed(12345)
N = 100000
exp_means <- numeric(N)
for (i in 1:N) {
  x <- 15 * rexp(25, 1) + 85 #transform so mean = 100, sd = 15
  exp_means[i] <- mean(x)
}
head(exp_means)
```

```
[1] 103.28723 100.75480 98.79687 97.14471 99.90882 95.07226
```

```
exp_means %>% hist(breaks=100,main="Smp. Dist. of exp Means")
```



```
plot(density(exp_means),main="Smp. Dist. of exp Means")
```



```
mean(exp_means)
```

```
[1] 100.004
```

```
sd(exp_means)
```

```
[1] 3.00933
```

This distribution has a longer tail on the right than on the left but is not as extreme as the lognormal distribution. The sd looks like it's near 3, just like the lognormal distribution above, though it was transformed.

Overall, what conclusions do you make about the applicability of the Central Limit Theorem given what we have demonstrated with variables from:

The binomial distribution (in class). The normal distribution (warm-up). The uniform distribution (warm-up). The lognormal (on your own). The exponential (on your own).

With enough iterations, sample distributions for each of the above distributions converge to the same distribution: the normal distribution.

Part 3: Problems from your peers!

You attend a party where there are already 20 guests in the room. Unbeknownst to you, 5 guests are zombies, and 7 are vampires.

One person approaches you and buys you a drink. What is the probability that this person is a vampire?

Assuming a vampire is equally as likely approach me and buy me a drink as any other guest, (Probably a naive assumption)

$$\begin{aligned} P(V) &= \frac{7}{20} \\ &= 0.35 \end{aligned}$$

Two people approach you and ask your opinion on the host's outfit. What is the probability that they are both zombies?

Assuming a zombie is equally as likely to approach me and ask my opinion on the host's outfit as any other guest and that these two events are independent, (Again, probably a naive assumption; it is a well known fact that zombies perform a complex flocking phenomenon.)

$$\begin{aligned} P(Z_1) &= \frac{5}{20} = 0.25 \\ P(Z_2) &= \frac{4}{19} \approx 0.21 \\ P(Z_1) \cap P(Z_2) &= P(Z_1)P(Z_2) \text{ from Def. 2.4.2 (p. 73)} \\ &\approx 0.25 \times 0.21 \\ &\approx 0.0525 \end{aligned}$$

Three people approach you and ask you to be the fourth player in their Texas hold'em game. What is the probability that they are all normal humans?

Assuming all the guests at the party who are not vampires or zombies are normal humans and assuming the aforementioned bits about equal likelihood and independence,

$$\begin{aligned} P(H_1) &= \frac{8}{20} = 0.4 \\ P(H_2) &= \frac{7}{19} \approx 0.368 \\ P(H_3) &= \frac{6}{18} \approx 0.333 \\ P(H_1) \cap P(H_2) \cap P(H_3) &= P(H_1)P(H_2)P(H_3) \text{ from Def. 2.4.2 (p. 73)} \\ &\approx 0.4 \times 0.368 \times 0.333 \\ &\approx 0.0490176 \end{aligned}$$

Bud only goes out trick-or-treating when there are clear skies (not too dark or too wet) and there is no full moon (he's superstitious). There is a full moon every 27.32 days. Assuming it is a random Halloween – i.e. we are not aware of any weather forecast or pattern, nor recent moon phases – and the probability of clear skies on a random October 31 is 0.6, what is the probability that Bud will go trick-or-treating?

M = full moon

C = clear skies

T = Bud goes out trick-or-treating

$$P(M) = \frac{1}{27.32} \approx 0.037$$

$$P(C) = .06(\text{given})$$

$$P(T) = P(C)P(M^c)$$

$$P(M^c) = 1 - P(M) \text{ from Property 1 (p. 58)}$$

$$P(T) = P(C)(1 - P(M))$$

$$P(T) \approx 0.6 \times (1 - 0.037)$$

$$\begin{aligned} P(T) &\approx 0.6 \times 0.963 \\ &\approx 0.5778 \end{aligned}$$