Spring 2020
GE 461 Introduction to Data Science

Statistical Models by Savaş Dayanık
Advertising and Promotion

Contents

1 Introduction

The Dodgers is a professional baseball team and plays in the Major Baseball League. The team owns a 56,000-seat stadium and is interested in increasing the attendance of their fans during home games. *At the moment the team management would like to know if bobblehead promotions increase the attendance of the team's fans?* This is a case study based on Miller (2014 Chapter 2).

```
include_graphics(c("los_angeles-dodgers-stadium.jpg",
                   "Los-Angeles-Dodgers-Promo.jpg",
                   "adrian_bobble.jpg"))
```

The 2012 season data in the `events` table of SQLite database `data/dodgers.sqlite` contain for each of 81 home play the

- month,
- day,
- weekday,
- part of the day (day or night),
- attendance,
- opponent,
- temperature,
- whether cap or shirt or bobblehead promotions were run, and
- whether fireworks were present.

Figure 1: 56,000-seat Dodgers (left), stadium (middle), shirts and caps (right) *bobblehead*

2 Prerequisites

We will use `R`, `RStudio`, `R Markdown` for the next three weeks to fit statistical models to various data and analyze them. Read Wickham and Grolemund (2017) online

- Section 1.1 for how to download and install `R` and `RStudio`,
- Chapter 27 for how to use `R Markdown` to interact with `R` and conduct various predictive analyses.

All materials for the next three weeks will be available on Google drive.

3 Exploratory data analysis

1. Connect to `data/dodgers.sqlite`. Read table `events` into a variable in `R`.

    - Read Baumer, Kaplan, and Horton (2017 Chapters 1, 4, 5, 12) for getting data from and writing them to various SQL databases.
    - Because we do not want to hassle with user permissions, we will use SQLite for practice. I recommend `PostgreSQL` for real projects.
    - Open `RStudio` terminal, connect to database `dodgers.sqlite` with `sqlite3`. Explore it (there is only one table, `events`, at this time) with commands
        - `.help`
        - `.databases`
        - `.tables`
        - `.schema <table_name>`
        - `.headers on`
        - `.mode column`
        - `SELECT ...`
        - `.quit`
    - Databases are great to store and retrieve large data, especially, when they are indexed with respect to variables/columns along with we do search and match extensively.
    - `R` (likewise, `Python`) allows one to seeminglessly read from and write to databases. For fast analysis, keep data in a database, index tables for fast retrieval, use `R` or `Python` to fit models to data.

```r
library(RSQLite)
con <- dbConnect(SQLite(), "../data/dodgers.sqlite")
# dbListTables(con)
```

```r
# this will let sqlite do all jobs for us
events <-  tbl(con, "events")

# pipes (%>%) below allow us to run chains of commands without having to
# creating temporary variables in between (R does that automatically
# for us)
events %>%
  select(month, day, day_of_week, opponent, bobblehead, attend) %>%
  head() %>%
  collect() %>%
  pander(caption = "A glimpse (first six rows and columns) of data retrieved from events table of d
```

Table 1: A glimpse (first six rows and columns) of data retrieved from events table of database

month	day	day_of_week	opponent	bobblehead	attend
APR	10	Tuesday	Pirates	NO	56000
APR	11	Wednesday	Pirates	NO	29729
APR	12	Thursday	Pirates	NO	28328
APR	13	Friday	Padres	NO	31601
APR	14	Saturday	Padres	NO	46549
APR	15	Sunday	Padres	NO	38359

```r
# Next command copies the entire data to the memory of local machine. Do not do
# this if the table is large

d <- dbReadTable(con, "events")
```

2. What are the number of plays on each week day and in each month of a year?

```r
# after class. Check the Rmd file for work done in class. Here I write the streamlined version add
d %>%
  mutate(day_of_week = factor(day_of_week, levels = c("Monday", "Tuesday", "Wednesday", "Thursday",
  count(day_of_week, name = "Number of games") %>%
  rename(`Week day`= day_of_week) %>%
  pander(caption = "Number of games on week days")
```

Table 2: Number of games on week days

Week day	Number of games
Monday	12
Tuesday	13
Wednesday	12
Thursday	5
Friday	13
Saturday	13
Sunday	13

The games were played pretty much uniformly across each week day except Thursday, which has less than half of the games than other days.

```r
d %>%
  mutate(month = factor(month, levels = c("APR", "MAY", "JUN", "JUL", "AUG", "SEP", "OCT"))) %>%
  count(month, name = "Number of games") %>%
  rename(`Month`= month) %>%
  pander(caption = "Number of games across months")
```

Table 3: Number of games across months

Month	Number of games
APR	12
MAY	18
JUN	9
JUL	12
AUG	15
SEP	12
OCT	3

May hosted the greatest number of games, while October the least. June has as much as the half of games in May. The remainder months have high and similar game numbers.

```r
# in class
d %>% dim() %>% `[` (1)
dim(d)[1]

d %>% dim()

d %>% count(day_of_week, sort=TRUE)
d %>% count(day_of_week, day_night, name = "cnt", sort=TRUE) %>%
  pivot_wider(names_from = day_night, values_from = cnt)

d %>% count(day_of_week, bobblehead, name = "cnt", sort=TRUE) %>%
  pivot_wider(names_from = bobblehead, values_from = cnt)

d %>%
  group_by(day_of_week) %>%
  summarize(mean = mean(attend)) %>%
  arrange(mean) %>%
  ggplot(aes(day_of_week, mean)) +
  geom_point()

d %>%
  count(month, sort=TRUE)
```

3. Check the orders of the levels of the `day_of_week` and `month` factors. If necessary, put them in the logical order.

```r
d2 <- d %>%
  mutate(day_of_week = factor(day_of_week, levels = c("Monday", "Tuesday", "Wednesday", "Thursday",
     month = factor(month, levels = c("APR", "MAY", "JUN", "JUL", "AUG", "SEP", "OCT")))
d2 %>%
  select(day_of_week, month) %>%
  summary() %>%
  pander(caption = "Month and week day names now follow time order.")
```

Table 5: Number of times booblehead was given away on games played different weekdays

	Bobblehead	
Weekday	NO	YES
Monday	12	.
Tuesday	7	6
Wednesday	12	.
Thursday	3	2
Friday	13	.
Saturday	11	2
Sunday	12	1

Table 4: Month and week day names now follow time order.

day_of_week	month
Monday :12	APR:12
Tuesday :13	MAY:18
Wednesday:12	JUN: 9
Thursday : 5	JUL:12
Friday :13	AUG:15
Saturday :13	SEP:12
Sunday :13	OCT: 3

4. How many times were bobblehead promotions run on each week day?

```
d2 %>%
  count(day_of_week, bobblehead, name = "cnt") %>%
  pivot_wider(names_from = bobblehead, values_from = cnt) %>%
  rename(`Weekday` = day_of_week) %>%
  kable(format = kable_format, caption = "Number of times booblehead was given away on games playe
  kable_styling(full_width = FALSE) %>%
  add_header_above(c(" "=1, "Bobblehead"=2))
```

Bobbleheads were given away in total 11 out of 81 games. Eight of bobbleheads were given during the weekdays, Tuesday and Thursdays Thuesday takes the leads with more than half of all bobbleheads given during season.

5. How did the attendance vary across week days? Draw boxplots. On which day of week was the attendance the highest on average?

```
d2 %>%
  ggplot(aes(day_of_week, attend, group=1)) +
  geom_point() +
  scale_y_continuous(labels=scales::comma) +
  geom_smooth(se=FALSE, method="loess")
```

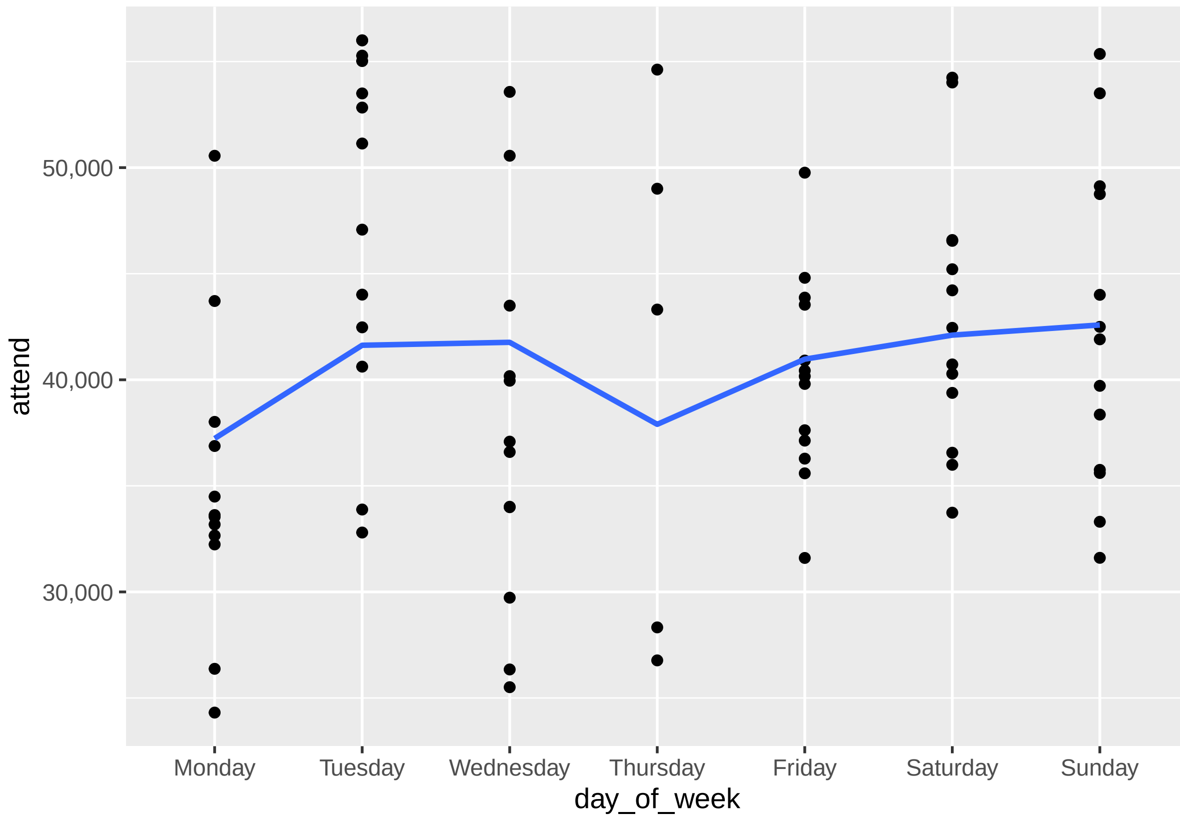


Figure shows 81 game attendance numbers with a loess smoother. The average attendance stays pretty much constant a little above 40,000. Only five games were played on Thursdays, so it is hard to say that attendance on Thursday games are decisively lower than average. However, Monday has more data and games on Monday tended to attract lower fans in the stadium.

6. Is there an association between attendance and
   - whether the game is played in day light or night?
   - Between attendance and whether skies are clear or cloudy?

```
Draw separate boxplots and comment on the plots.
```

```
d2 %>%
  ggplot(aes(day_night, attend)) +
  geom_boxplot(aes(fill=day_night)) +
  theme(legend.position = "none")

d2 %>%
  ggplot(aes(skies, attend)) +
  geom_boxplot(aes(fill=skies)) +
  theme(legend.position = "none")
```

```
We can run a formaly Chi-suared test of independence.
```

```
skies_tbl <- d2 %>%
      mutate(attend_cut = cut(attend, breaks = c(0, quantile(attend, prob=(1:2)/3), Inf))) %>%
      xtabs(~ attend_cut + skies, .)

skies_tbl %>%
  as_tibble() %>%
```

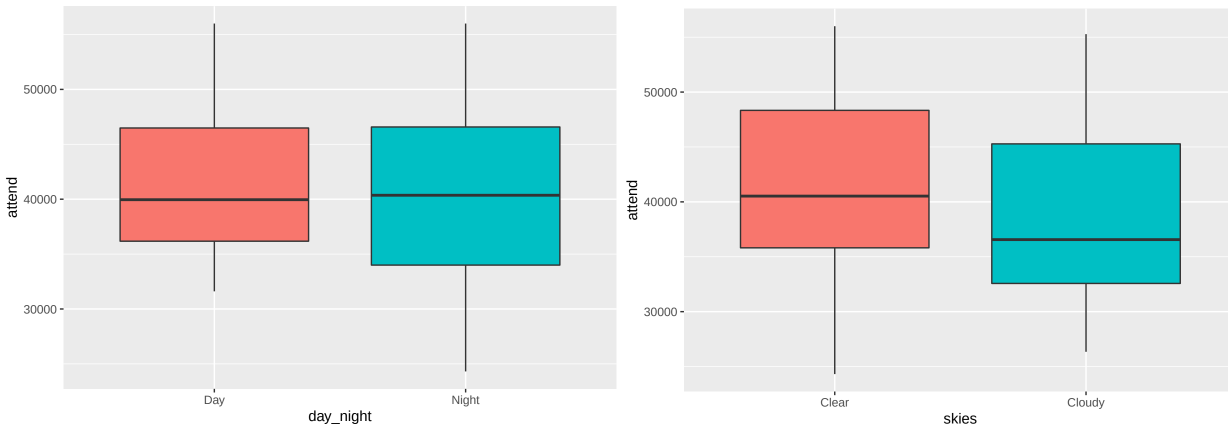


Figure 2: Attendance distributions across games played in day light and at night are displayed on the left. Medians are close, and mid 50% coincide. Game time does not seem to be marginally important. On the right, the median attendances for games played under clear and cloudy skies look different, but the difference does not seem significant when the variations of mid 50% attendance numbers are taken into account. Those variations can reduce and difference can stick out after we take other explanatory variables into account.

Table 6: Note that more than half of the games played under cloudy skies have attendance on the lower side, whereas only less than one third of the games played under clear skies are on the lower side.

	skies	
attend_cut	Clear	Cloudy
(0,3.66e+04]	17	10
(3.66e+04,4.39e+04]	24	3
(4.39e+04,Inf]	21	6

```
  pivot_wider(names_from=skies, values_from = n) %>%
  kable(caption = "Note that more than half of the games played under cloudy skies have attendance on t
  kable_styling(full_width = FALSE) %>%
  add_header_above(c(" "= 1, "skies" = 2))
```

```
chisq.test(skies_tbl) %>%   pander()
```

Table 7: Pearson's Chi-squared test: `skies_tbl`

Test statistic	df	P value
5.088	2	0.07854

Chi-square test has a marginal 7% p-value, under which we cannot reject independence of attendance and skies, but its small value calls for a more comprehensive analysis together with other values.

7. Is there an association between attendance and temperature?
   - If yes, is there a positive or negative association?
   - Do the associations differ on clear and cloud days or day or night times?

   Draw scatterplots and comment.

```
d2 %>%
  ggplot(aes(temp, attend)) +
  geom_point() +
  geom_smooth(se=FALSE, method="loess")
```

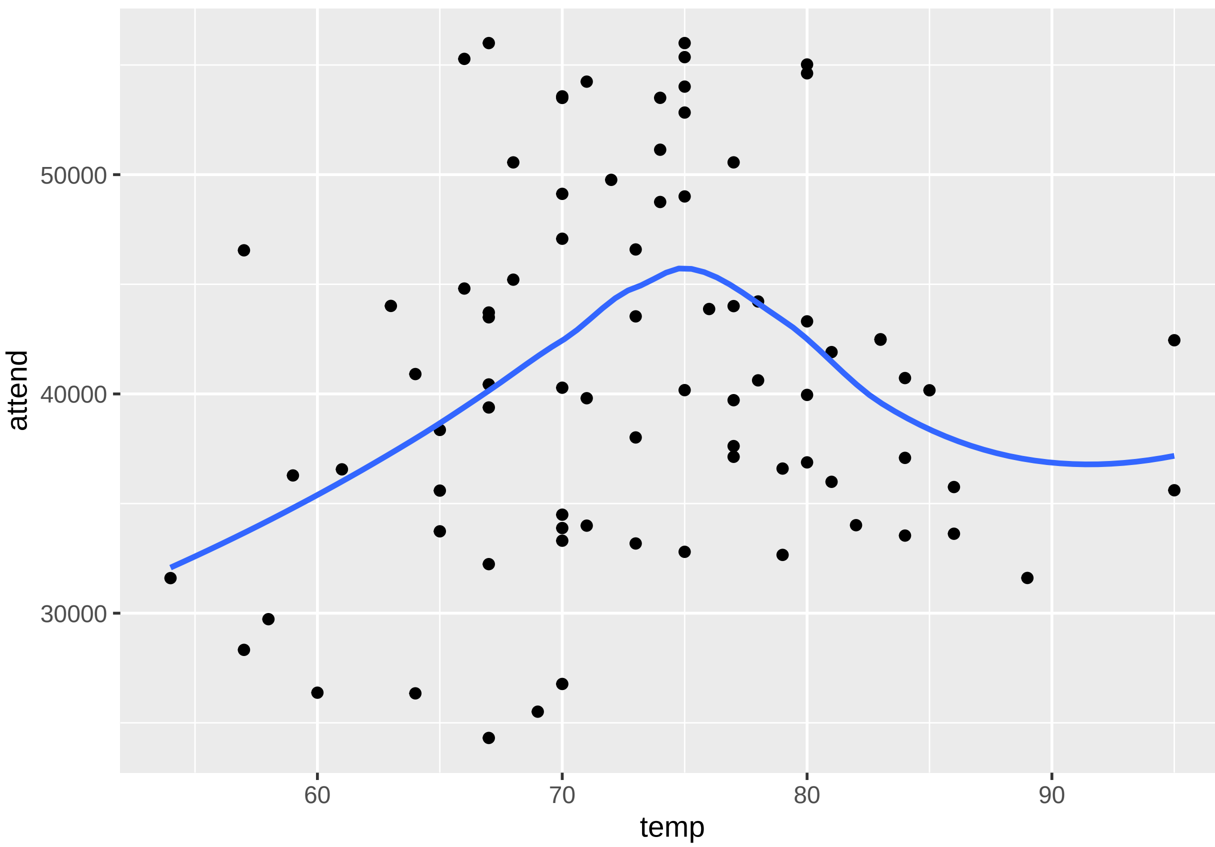


Figure 3: The smoother makes clear that uncomfortably low and high temperatures discourage fans from attending game in the stadium.

4 A linear regression model

Regress attendance on month, day of the week, and bobblehead promotion.

$$\text{attendance}_i = \beta_0 + \beta_{MAY}\delta_{MA,i} + \ldots + \beta_{OCT}\delta_{OCT,i} + \beta_{Tue}\delta_{Tue,i} + \ldots + \beta_{Sun}\delta_{Sun,i} + \beta_{YES}\delta_{YES,i} + \varepsilon_i.$$

for $i = 1, \ldots, 81$, where $\varepsilon_i \sim \text{Normal}(0, \sigma^2)$, and the δ are dummy variables, one if the associated event occurs for the ith game, and zero otherwise.

$$\begin{aligned}
\beta_0 &: \text{average attendance for a typical game played on some Monday in APR when no bobblehead was NOT given away,}\\
\beta_{MAY} &: \text{average difference in attendance for a typical game played in MAY rather than APR,}\\
\beta_{Tue} &: \text{average difference in attendance for a typical game played on Tuesday rather than Monday,}\\
\beta_{YES} &: \text{average difference in attendance for a typical game when a bobblehead is given away.}
\end{aligned}$$

and other betas are defined similarly.

We find the β with maximum likelihood:

```r
lmod <- lm(attend ~ month + day_of_week + bobblehead, d2)
lmod
```

```

Call:
lm(formula = attend ~ month + day_of_week + bobblehead, data = d2)

Coefficients:
         (Intercept)              monthMAY              monthJUN
            33909.16              -2385.62               7163.23
            monthJUL              monthAUG              monthSEP
             2849.83               2377.92                 29.03
            monthOCT    day_of_weekTuesday  day_of_weekWednesday
             -662.67               7911.49               2460.02
 day_of_weekThursday     day_of_weekFriday   day_of_weekSaturday
              775.36               4883.82               6372.06
   day_of_weekSunday         bobbleheadYES
             6724.00              10714.90
```

```r
# lmod %>% pander(caption = "Linear regression model")
```

- We expect 33,909 attendance on a game played on some Monday in APR and no bobblehead was given.
- If, instead, game is played on MAY, the attendance is expected to drop by 2,386.
- If a bobblehead is given away, then we expect attendance to increase by 10,715.

The bobblehead seems to increase the attendance number by a larger quantity than any other factor. However, is the difference statistically significant? We will test it below.

8. Is there any evidence for a relationship between attendance and other variables? Why or why not?

```r
small <- update(lmod, . ~ 1 )
anova(small, lmod)
```

```
Analysis of Variance Table

Model 1: attend ~ 1
Model 2: attend ~ month + day_of_week + bobblehead
  Res.Df        RSS Df  Sum of Sq      F       Pr(>F)
1     80 5507932888
```

```
2     67 2509574563 13 2998358324 6.1576 0.0000002083 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Test $H_0 : \beta_{MAY} = \ldots = \beta_{OCT} = \beta_{Tue} = \ldots = \beta_{Sun} = \beta_{YES} = 0$

We reject small/null model because F stat is large (or p-value is small <= 0.05). We conclude that at least one of variables on thr right has some reltion to attendance.

9. Does the bobblehead promotion have a statistically significant effect on the attendance?

Test $H_0 : \beta_{YES} = 0$.

```r
small <-  update(lmod, . ~ . - bobblehead)
anova(small, lmod)
```

```
Analysis of Variance Table

Model 1: attend ~ month + day_of_week
Model 2: attend ~ month + day_of_week + bobblehead
  Res.Df        RSS Df Sum of Sq      F    Pr(>F)
1     68 3244161740
2     67 2509574563  1 734587177 19.612 0.0000359 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Since p-value is practically zero, we reject the small model, which means that bobblehead is important
```

10. Do month and day of week variables help to explain the number of attendants?

Similarly, we will conduct F tests.

```r
drop1(lmod, test="F")
```

```
Single term deletions

Model:
attend ~ month + day_of_week + bobblehead
            Df Sum of Sq        RSS    AIC F value    Pr(>F)
<none>                   2509574563 1425.2
month        6 620147363 3129721926 1431.0  2.7594   0.01858 *
day_of_week  6 575839199 3085413762 1429.9  2.5623   0.02704 *
bobblehead   1 734587177 3244161740 1444.0 19.6118 0.0000359 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

All explanatory variables are needed in the model to get a good accuracy on future predictions (AIC) and to explain variation in the existing attendance data (F-test). So we stick to full model.

$$AIC = -2 \times \text{log-likelihood} + 2 \times \text{number of parameters in the model}$$

11. How many fans are expected to be drawn **alone by a bobblehead promotion** to a home game? Give a 90% confidence interval.

The expected additional number of attendance is β_{YES}, which is estimated as 10,715.

```r
confint(lmod, level = 0.90)["bobbleheadYES",]
```

```
      5 %      95 %
 6679.347 14750.460
```

12. How good does the model fit to the data? Why? Comment on residual standard error and R^2. Plot observed attendance against predicted attendance.

R2 is the fraction of variation in attendance (in the past 81 games) explained by the current (month, day_of_week, bobblehead). Here, R2 becomes 54% of variation explained. Our model is not too bad, but not too good either.

The standard deviation of error is 6,120, which is 15% of average attendance we expect. Compared to 40% typical error percentage, this is not bad.

13. Predict the number of attendees to a typical home game on a **Wednesday** in **June** if a **bobblehead promotion** is extended. Give a 90% prediction interval.

```
d2$month %>% levels()
```

```
[1] "APR" "MAY" "JUN" "JUL" "AUG" "SEP" "OCT"
```

```
d2$day_of_week %>%  levels()
```

```
[1] "Monday"    "Tuesday"   "Wednesday" "Thursday"  "Friday"    "Saturday"
[7] "Sunday"
```

```
d2$bobblehead %>%  unique()
```

```
[1] "NO"  "YES"
```

```
newdata <- data.frame(month = "JUN", day_of_week = "Wednesday",
                      bobblehead = "YES")
predict(lmod, newdata=newdata, level=0.90,
        interval = "prediction")
```

```
       fit     lwr      upr
1 54247.32 42491.1 66003.55
```

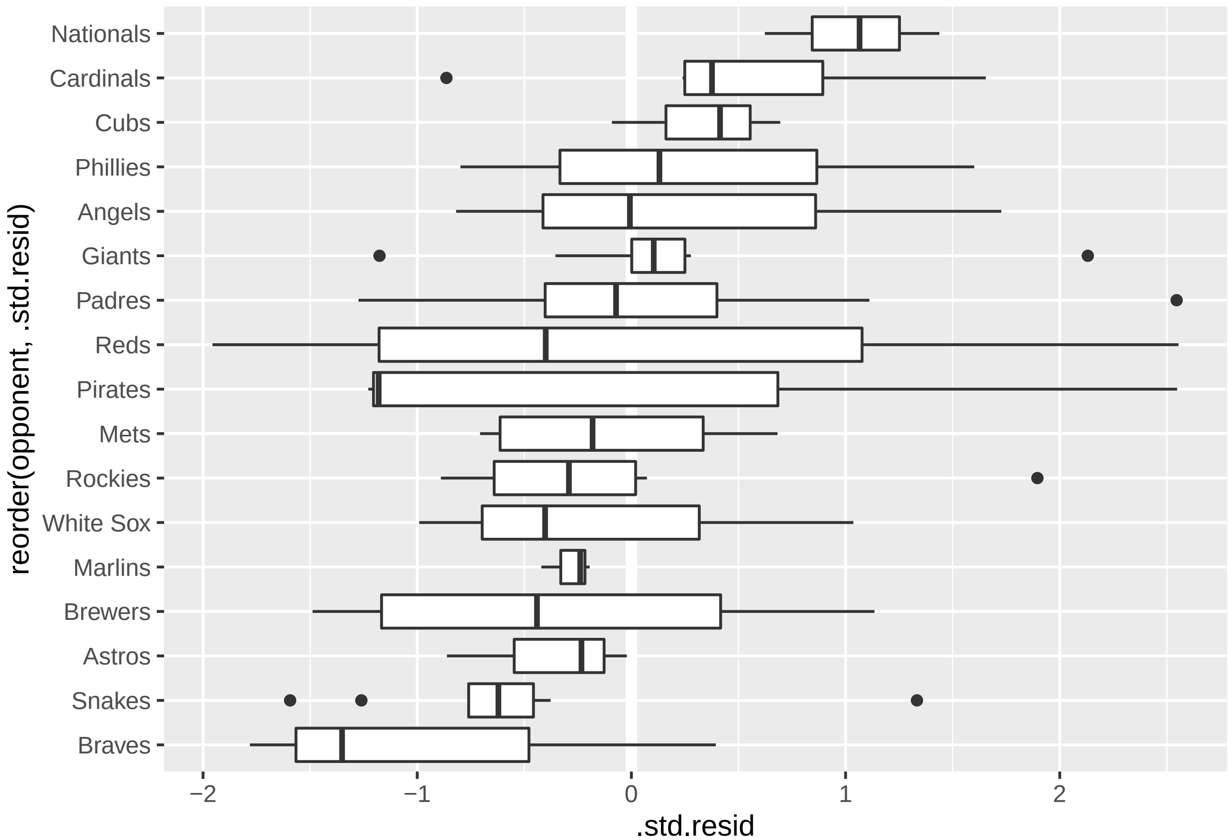
```
# ?predict.lm
```

5 More ideas about improving our model

5.1 Introduce new variables.

Let us check if `opponent` variable should be added to the model.

```
broom::augment(lmod, data=d2) %>%
  mutate(opponent = factor(opponent)) %>%
  ggplot(aes(reorder(opponent, .std.resid), .std.resid)) +
  geom_hline(yintercept=0, col="white", size=2) +
  geom_boxplot() +
  coord_flip()
```



Because we are consistently under- and over-estimating attendance in games played against certain opponents, it is a good idea to add. opponent.

5.2 Introduce interaction between existing variables

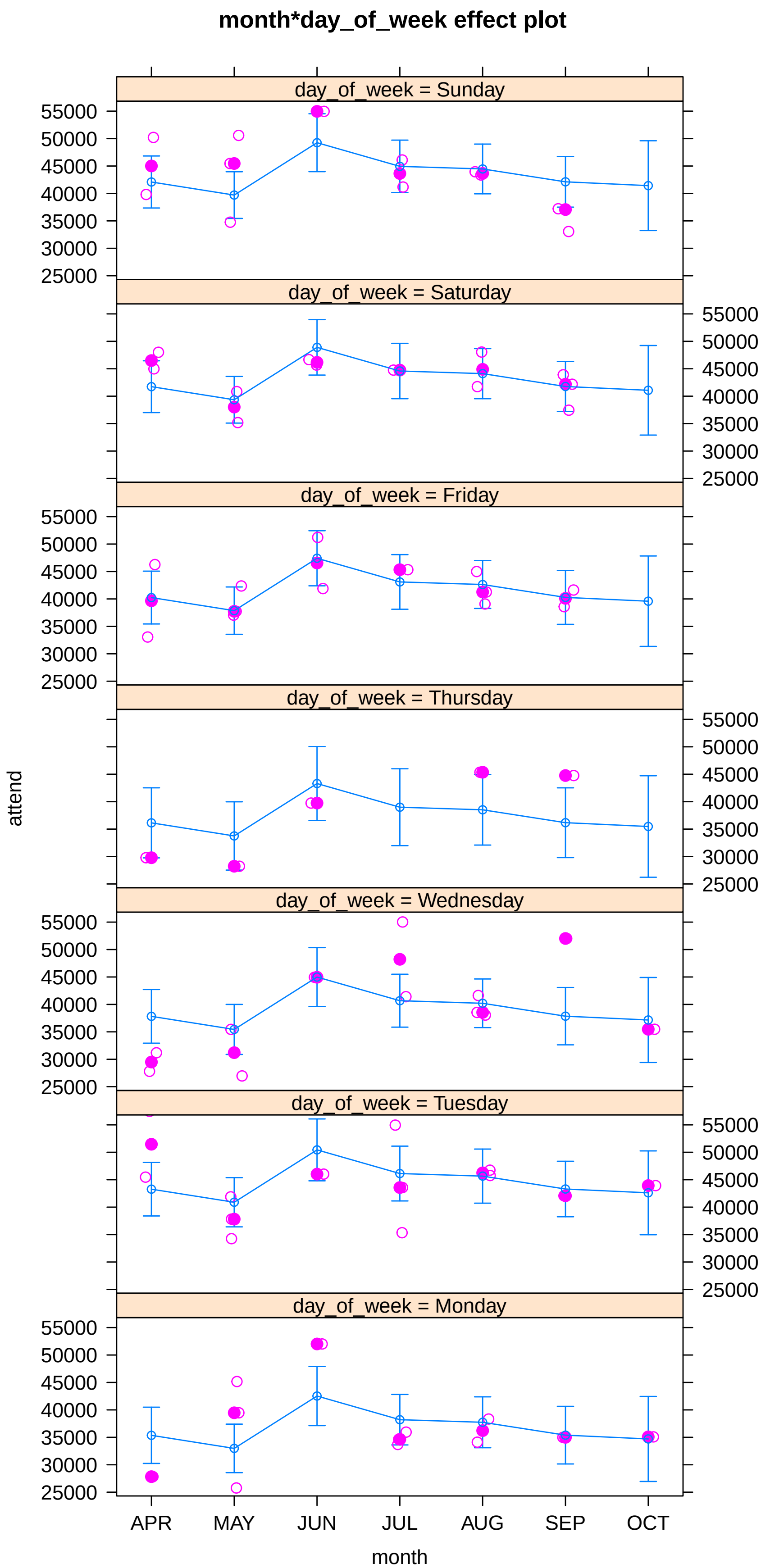
It is natural to expect interaction between month and day_of_week: during summer months, people are expected to spend more time outdoors, especially, during weekends. Let us see if data are consistent with current no-interaction model estimates.

```
d3 <- mutate(d2, bobblehead = factor(bobblehead))
lmod <- update(lmod, data  = d3)

effects::effect(c("month","day_of_week"), lmod,
                partial.residuals = TRUE ) %>% plot(layout = c(1,7))
```

```
## Warning in term == terms: longer object length is not a multiple of shorter
## object length

## Warning in term == names: longer object length is not a multiple of shorter
## object length
```


month*day_of_week effect plot
day_of_week = Sunday
day_of_week = Saturday
day_of_week = Friday
day_of_week = Thursday
day_of_week = Wednesday
day_of_week = Tuesday
day_of_week = Monday
55000
50000
45000
40000
35000
30000
25000
attend
APR
MAY
JUN
JUL
AUG
SEP
OCT
month

Model seems to overestimate effects of Wednesday and Thursday in APR and MAY and underestimate Tursday effects in AUG and SEP. Let us augment model with interaction term and run an F-test to check if it is statistically significant.

5.2.1 Model selection with F-test

```
lmod
```

```
Call:
lm(formula = attend ~ month + day_of_week + bobblehead, data = d3)

Coefficients:
         (Intercept)              monthMAY              monthJUN
            33909.16              -2385.62               7163.23
            monthJUL              monthAUG              monthSEP
             2849.83               2377.92                 29.03
            monthOCT    day_of_weekTuesday  day_of_weekWednesday
             -662.67               7911.49               2460.02
 day_of_weekThursday     day_of_weekFriday   day_of_weekSaturday
              775.36               4883.82               6372.06
   day_of_weekSunday         bobbleheadYES
             6724.00              10714.90
```

```
large <- update(lmod, . ~ . + month:day_of_week)
anova(lmod, large)
```

```
Analysis of Variance Table

Model 1: attend ~ month + day_of_week + bobblehead
Model 2: attend ~ month + day_of_week + bobblehead + month:day_of_week
  Res.Df        RSS Df  Sum of Sq    F Pr(>F)
1     67 2509574563
2     36 1082895916 31 1426678647 1.53 0.1096
```

P-value is large, so we cannot reject small model. Therefore, F-test concluded that interaction between month and day_of_week is unimportant.

5.2.2 Model selection with AIC

```
 final <- update(lmod, . ~ .^2) %>%  step()
```

```
Start:  AIC=1422.53
attend ~ month + day_of_week + bobblehead + month:day_of_week +
    month:bobblehead + day_of_week:bobblehead

                         Df  Sum of Sq        RSS    AIC
- day_of_week:bobblehead  1   12082574 1061414706 1421.5
- month:bobblehead        1   17121963 1066454095 1421.8
<none>                                 1049332132 1422.5
- month:day_of_week      27 1335988226 2385320358 1435.0

Step:  AIC=1421.46
attend ~ month + day_of_week + bobblehead + month:day_of_week +
    month:bobblehead
```

```
                     Df  Sum of Sq        RSS    AIC
- month:bobblehead    2   21481210 1082895916 1419.1
<none>                             1061414706 1421.5
- month:day_of_week 29 1363309764 2424724470 1430.4

Step:  AIC=1419.08
attend ~ month + day_of_week + bobblehead + month:day_of_week

                     Df  Sum of Sq        RSS    AIC
<none>                             1082895916 1419.1
- month:day_of_week 31 1426678647 2509574563 1425.2
- bobblehead          1  351400539 1434296455 1439.8
```

Step() applies repeatedly drop1 to find the model with the least AIC until no new term is found to drop out of last model. So interaction terms other than month:day_of_week turn out be unimportant from prediction accuracy on unseen data as estimated by AIC.

5.2.3 Model selection with cross-validation and t-test

Check if the interaction term is necessary with **cross-validation**

```r
set.seed(461)

nfolds <- 5
folds <- rep(seq(5), nrow(d2), len=nrow(d2)) %>% sample()
rmse_lmod <- rep(NA, nfolds)
rmse_lmod_interaction <- rep(NA, nfolds)

lmod_interaction <- update(lmod, . ~ . + month:day_of_week)

for (i in seq(nfolds)){
  train <- d2[folds!=i,]
  test <- d2[folds==i,]

  # train lm without interaction model
  lmod_train <- update(lmod, data = train)
  lmod_test <- predict(lmod_train, newdata = test)
  rmse_lmod[i] <- (test$attend - lmod_test)^2 %>% mean() %>% sqrt()

  # train lm with interaction model
  lmod_interaction_train <- update(lmod_interaction, data = train)
  lmod_interaction_test <- suppressWarnings(predict(lmod_interaction_train, newdata = test))
  rmse_lmod_interaction[i] <- (test$attend - lmod_interaction_test)^2 %>% mean() %>% sqrt()
}

cv <- tibble(lmod = rmse_lmod, lmod_interaction = rmse_lmod_interaction) %>%
  mutate(dif_rmse = lmod - lmod_interaction)

cv %>%
  apply(2,mean)
```

```
            lmod lmod_interaction         dif_rmse
        6582.020         9640.356        -3058.336
```

```r
p1 <- cv %>%
  pivot_longer(cols=c(lmod, lmod_interaction), names_to = "model", values_to = "rmse") %>%
```

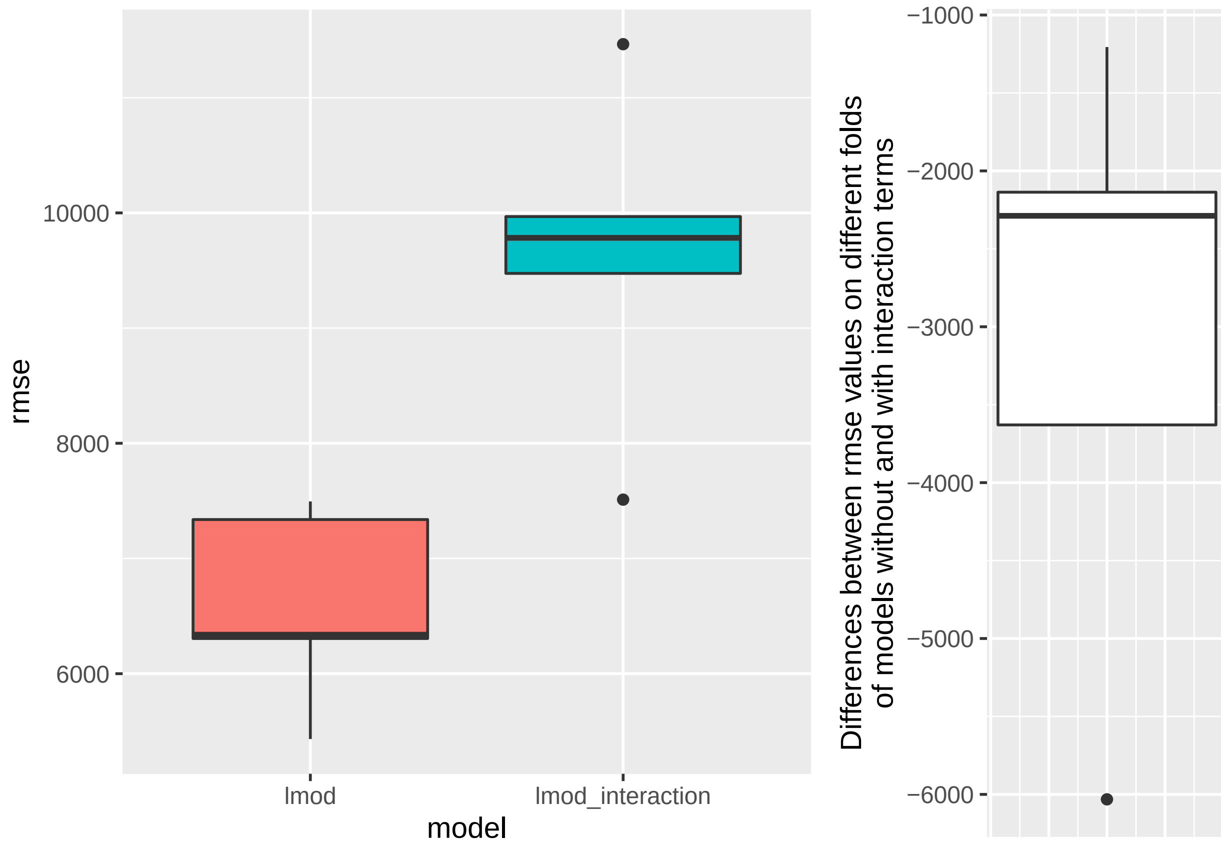


Figure 4: Comparison of models without and with interaction term

```
  ggplot(aes(model, rmse)) +
  geom_boxplot(aes(fill=model)) +
  theme(legend.position = "none")

p2 <- cv %>%
  ggplot(aes(1, dif_rmse)) +
  geom_boxplot() +
  labs(y = "Differences between rmse values on different folds\nof models without and with interaction t
  theme(axis.text.x = element_blank(), axis.ticks.x = element_blank())

gridExtra::grid.arrange(p1,p2,layout_matrix = matrix(c(1,1,2), nrow=1))
```

Run a **two-sided t-test** on the difference of rmse values to check if the mean rmse values for models with and without interaction terms are the same.

```
t.test(x = cv$dif_rmse)
```

```
    One Sample t-test

data:  cv$dif_rmse
t = -3.6498, df = 4, p-value = 0.02178
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -5384.8425  -731.8292
```

```
sample estimates:
mean of x
-3058.336
```

Because p-value is small, we conclude that the models have different mean rmse values. Because the t statistic is negative, model without interaction term seems to have a lower rmse than the model with interaction term.

6 Next: Nonlinear and nonparametric regression: recursive partitioning and random forests

Thursday lecture was cancelled as part of Covid 19 counter-measures.

7 Project (will be graded)

Include **all variables** and conduct a full regression analysis of the problem. Submit your `R markdown` and `html` files to course homepage on moodle in a **single** zip file. **Zip file must be named as "StuID".zip; for example, 21601224.zip** Below are the recap of the major steps to guide you through your project:

1. Explore the relation between attendance and explanatory variables with scatter, bar, box plots
2. Use Chi-squared test to check if there is attendance and explanatory variables are mutually independent.
3. Fit regression model with all explanatory variables.
    a. How good does your model fit to data?
    b. Which variables are significant? Use F-test, AIC, and cross-validation.
    c. Is bobblehead still significant? What is expected additional attendance due bobblehead? What is 80% confidence interval?
    d. Check model diagnostics.
        i. Does any quantitative explanatory variable need a nonlinear transformation?
        ii. Does your model benefit from adding two-way interaction terms between any two explanatory variables? Use cross-validation to support your decision.

Bibliography

Baumer, B.S., D.T. Kaplan, and N.J. Horton. 2017. *Modern Data Science with R*. Chapman & Hall/Crc Texts in Statistical Science. CRC Press. https://books.google.com.tr/books?id=NrddDgAAQBAJ.

Miller, T.W. 2014. *Modeling Techniques in Predictive Analytics with Python and R: A Guide to Data Science*. FT Press Analytics. Pearson Education. https://books.google.com.tr/books?id=PU6nBAAAQBAJ.

Wickham, H., and G. Grolemund. 2017. *R for Data Science*. O'Reilly Media. https://books.google.com.tr/books?id=aZRYrgEACAAJ.