Linear Models Marked Practical

P151, 1959 words

1 Exploratory Data Analysis

The data consist of 446 observations of the 6 variables described in the task. The main variable of interest, *time*, ranges from 21.1 to 278.06 seconds. Distances range from 50 to 400 meters; there are 224 and 222 observations, respectively, for women's and men's events. 191 events took place on long courses, i.e. at the Olympics, the remaining 255 on short courses at the World Championships. There are 80 observations each for Back- and Breaststroke, 79 each for Butterfly and Medley, and 128 for Freestyle events. The data set is thus well-balanced across each of the categories.

Figures 1 to 3 below shows the distributions of *time* for each stroke, segmented by whether the event was held on a short or a long course. The left panel of figure 1 shows distributions of times for men and women that are remarkably similar in shape for each stroke, although the times for men are consistently lower. The right panel shows that events held on short courses tend to see lower (i.e. better) times for all strokes except freestyle. This makes sense, as freestyle is the fastest of all 4 strokes, and it is also the stroke in which the effect of turns is the smallest. In Breaststroke and Butterfly, on the other hand, completing a 50m event on a 25m course means that a swimmer will have to turn once, which yields a time advantage, as the underwater phase that immediately follows a turn is typically quicker than the actual stroke in Breaststroke and Butterfly. This effect is less pronounced in Freestyle, as it is a faster stroke overall.

Figure 1: Distribution of *Time* across sex and course length

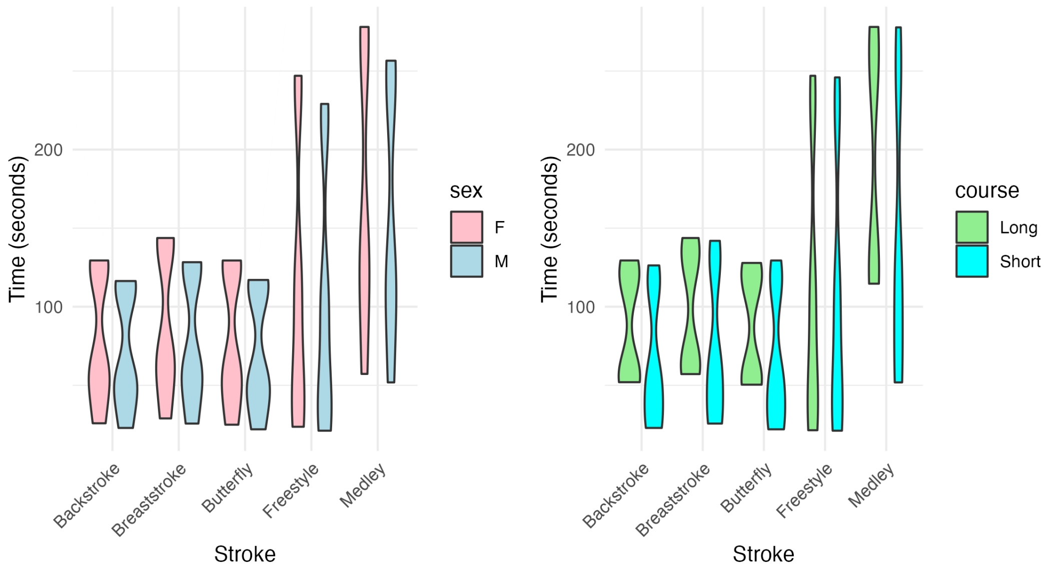
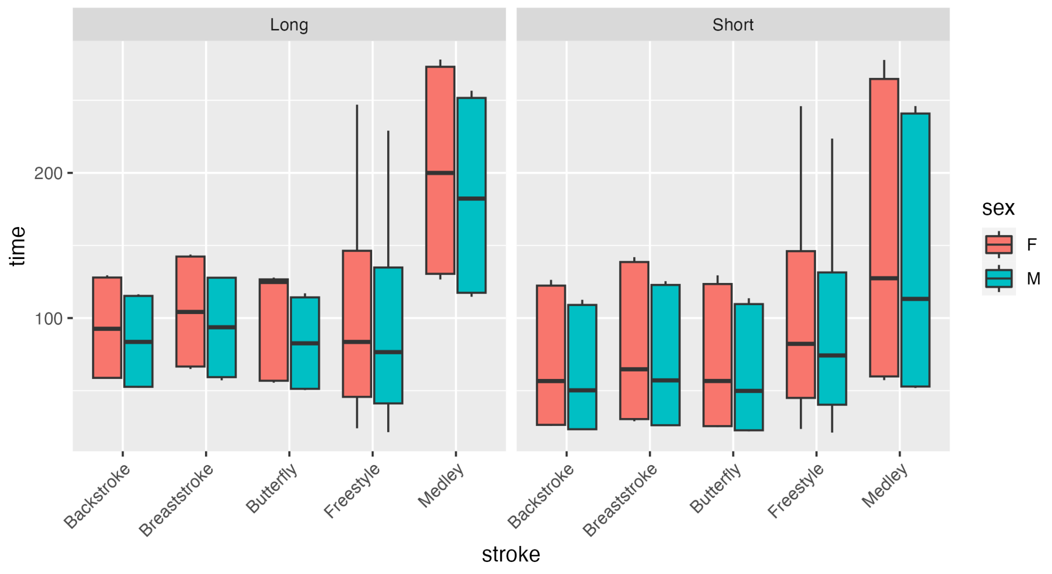


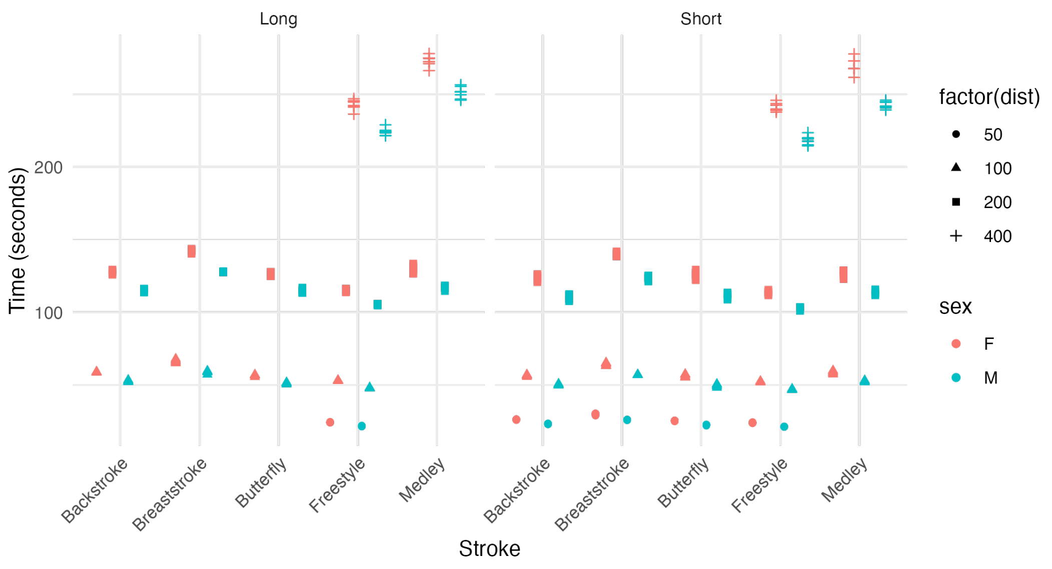
Figure 2 below shows that mean times across strokes and courses are lower for men than for women. This discrepancy is least pronounced in freestyle, and less pronounced on short courses than on long courses. The interquartile range for men's times is also marginally shorter than for women, suggesting slightly lower variation in times for male swimmers.

Figure 2: Time segmented by Sex, Course, and Stroke



Finally, Figure 3 below visualizes the distribution of times across the varying distances in addition to segmenting by sex and course length. We can see that the differences between male and female swimmers, as well as that between a short and a long course, hardly matter for short distance events (50 and 100m), but that they do become more pronounced for 200 and 400m events.

Figure 3: Time segmented by Sex, Course, and Distance



2 Data Preprocessing and the Baseline Model

Before attempting to model *time*, we first have to address two questions regarding our data. The first concerns the only variable not mentioned thus far, *event*, which at first glance looks to be a simple concatenation of *distance* and *stroke*. Upon closer inspection, however, we can see that *event* has 16 unique values (or levels, if we use it as a factor). This is problematic for two reasons: there are $4 \times 5 = 20$ distinct combinations of distance and stroke, which means that 4 of them are not present in our data. If our only aim was to do inference, this might be fine, but as we will attempt to do prediction later on, using the variable *event* will only work if all events we are trying to predict are in our data set. 400m Butterfly is not, so if we were to include *event* in our model, we could not use it to predict the time of such a race (which happens to be `RaceC` of the test data).

The fact that certain combinations of *distance* and *stroke* are not present in our data turns out to make a difference when fitting a linear model: Table 2 below shows R^2, $\bar{R}^2$, Residual SE and F-statistics for the following three linear models that regress *time* on all predictors (estimated coefficients omitted in the name of brevity):

$$time \sim dist * stroke + sex + course \tag{1}$$

$$time \sim dist + sex + course + stroke \tag{2}$$

$$time \sim dist + sex + course + stroke + \textbf{event}$$

We can see that simply excluding *event* results in a (very slightly) worse fit, higher Residual SE, and lower model complexity. Using *event* achieves the lowest residual SE, but instead replacing *event* with $dist * stroke$ saves us a few degrees of freedom, yields a higher F-statistic, and equal $\bar{R}^2$. Because we cannot use *event* to predict, and because we can replace it with an interaction of distance and stroke, I drop *event*.

Table 1: Inclusion of 'event' variable

	Dependent variable:		
	time		
	Without 'event'	With 'event'	with stroke * dist
	(1)	(2)	(3)
Observations	446	446	446
R^2	0.994	0.997	0.997
Adjusted R^2	0.994	0.997	0.997
Residual Std. Error	5.368 (df = 438)	3.681 (df = 428)	3.857 (df = 434)
F Statistic	10,767.190*** (df = 7; 438)	9,459.054*** (df = 17; 428)	13,305.300*** (df = 11; 434)
Note:			*p<0.1; **p<0.05; ***p<0.01

I use the model corresponding to Equation 1 as my baseline model moving forward.

The second question we have to ask ourselves is whether to use *distance* as a numeric or factor variable. Intuitively, meters swum is obviously numeric, but using it as such presents one problem in building our model: race distances are fixed and "increasing by 1 unit" does not have a sensible ceteris paribus interpretation. As this variable only has four distinct values, we can instead cast it as a factor variable. Table 2 compares Equation 1 for both data configurations (estimated coefficients again omitted). We can see that casting the distance as a factor increases the number of parameters, but also reduces residual SE. On balance, using *dist* as a categorical variable yields a lower AIC, so I cast *dist* as a factor.

Table 2: Full Model with 'dist' cast as

	Dependent variable:	
	time	
	Numeric	Factor
	(1)	(2)
Observations	446	446
R^2	0.997	0.997
Adjusted R^2	0.997	0.997
Akaike Inf. Crit.	2,483.733	2,447.678
Residual Std. Error	3.857 (df = 434)	3.681 (df = 428)
F Statistic	13,305.300*** (df = 11; 434)	9,459.054*** (df = 17; 428)
Note:		*p<0.1; **p<0.05; ***p<0.01

Finally, it appears sensible to investigate whether the large number of parameters used in the baseline model (1) might yield a worse fit than simply using the information contained in *dist* and *stroke* in an additive fashion, such as in (2). Table 2 below shows that the gains in parsimony (8 fewer parameters) from including these variables additively do not translate to a better overall fit. Model 1 has higher $\bar{R}^2$, lower AIC, lower residual SE and a higher F-statistic. I consider this sufficient justification to use it as a baseline.

Table 3: Baseline: 'dist' and 'stroke' included as

	Dependent variable:	
	time	
	Interaction	Additive
	(1)	(2)
Observations	446	446
R^2	0.997	0.995
Adjusted R^2	0.997	0.995
Akaike Inf. Crit.	2,447.678	2,745.651
Residual Std. Error	3.681 (df = 428)	5.185 (df = 436)
F Statistic	9,459.054*** (df = 17; 428)	8,978.044*** (df = 9; 436)
Note:		*p<0.1; **p<0.05; ***p<0.01

3 Model Improvement and Diagnostics

With a baseline model established, I now consider several possible ways of improving it. Since (1) includes all variables, it is possible that removing variables could yield improvement. However, automated backward selection based on both AIC and BIC shows that this is not the case: as the code snippet below shows, both information criteria deteriorate when we remove variables.

```
step_aic <- stepAIC(mod1, direction = "backward")
Start:  AIC=1179.98
time ~ dist * stroke + sex + course

              Df Sum of Sq     RSS    AIC
<none>                      5798.2 1180.0
- course       1     674.5  6472.7 1227.1
- dist:stroke  8    5924.5 11722.7 1478.0
- sex          1   12006.2 17804.4 1678.3

> step_bic <- step(mod1, direction = "backward", k = log(nrow(swim)))
Start:  AIC=1253.79
time ~ dist * stroke + sex + course

              Df Sum of Sq     RSS    AIC
<none>                      5798.2 1253.8
- course       1     674.5  6472.7 1296.8
- dist:stroke  8    5924.5 11722.7 1519.0
- sex          1   12006.2 17804.4 1748.1
```

Given that including *distance* and *stroke* as an interaction results in a better model than (2), it is possible that including additional interactions improves the model.

As we can see in Figure 2, Male and female swimmers have different times for the same strokes. For example, in freestyle and medley (both short and long), females tend to have slightly longer times than males. To see whether modelling this by means of an interaction term improves the model, I fit (3) below.

From Figure 3, we can see that for certain strokes, the time differences between males and females appear more pronounced in long distances compared to short distances. To incorporate this, I fit (4) below.

Lastly, we can try to incorporate the role of turns briefly discussed above by creating a new variable that captures the number of lengths swum in an event:

```
# create new variable that captures lengths swum
swim$lengths <- ifelse(swim$course == "Short",
                       as.numeric(as.character(swim$dist)) / 25,
                       as.numeric(as.character(swim$dist)) / 50)
```

Using this variable, I fit (5) and (6) below.

$$time \sim sex * stroke + dist + course \quad (3)$$

$$time \sim sex * dist + course + stroke \quad (4)$$

$$time \sim sex * stroke + lengths \quad (5)$$

$$time \sim dist * stroke + lengths * sex \quad (6)$$

Since these interaction models have many parameters, I will not produce regression tables with the results of each model here. Instead, Table 4 below displays the number of parameters, Akaike and Bayesian Information Criteria for each model. Models (3) - (5) are considerably more parsimonious than the baseline model, but have worse information criteria. Model (6) achieves the lowest AIC and BIC, but has a large number of parameters. I will restrict my attention to models 4 and 6. Model 4 achieves an AIC/BIC that is only marginally worse than the baseline while being more easily interpretable. Model 6 is unwieldy but results in a very good fit.

Table 4: Information Criteria Comparison

	Model 1	**Model 2**	**Model 3**	**Model 4**	**Model 5**	**Model 6**
p	22.00	10.00	14.00	13.00	11.00	23.00
aic	2447.68	2745.65	2721.51	2552.41	4525.56	1983.25
bic	2525.58	2790.75	2783.02	2609.82	4574.76	2065.26

Table 5 below displays estimated coefficients of both model 4 and model 6. The gaps in the right table represent combinations of *distance* and *stroke* that do not have a numerical point estimate. This can be the result of two things: non-existence of values (such as for 400m Butterfly events, as discussed earlier), or (perfect) multicollinearity. The number of singular coefficients suggests that model 6 is overfitting and should not be used for influence. It also prevents us from using it to predict later on.

Table 5: Regression Results

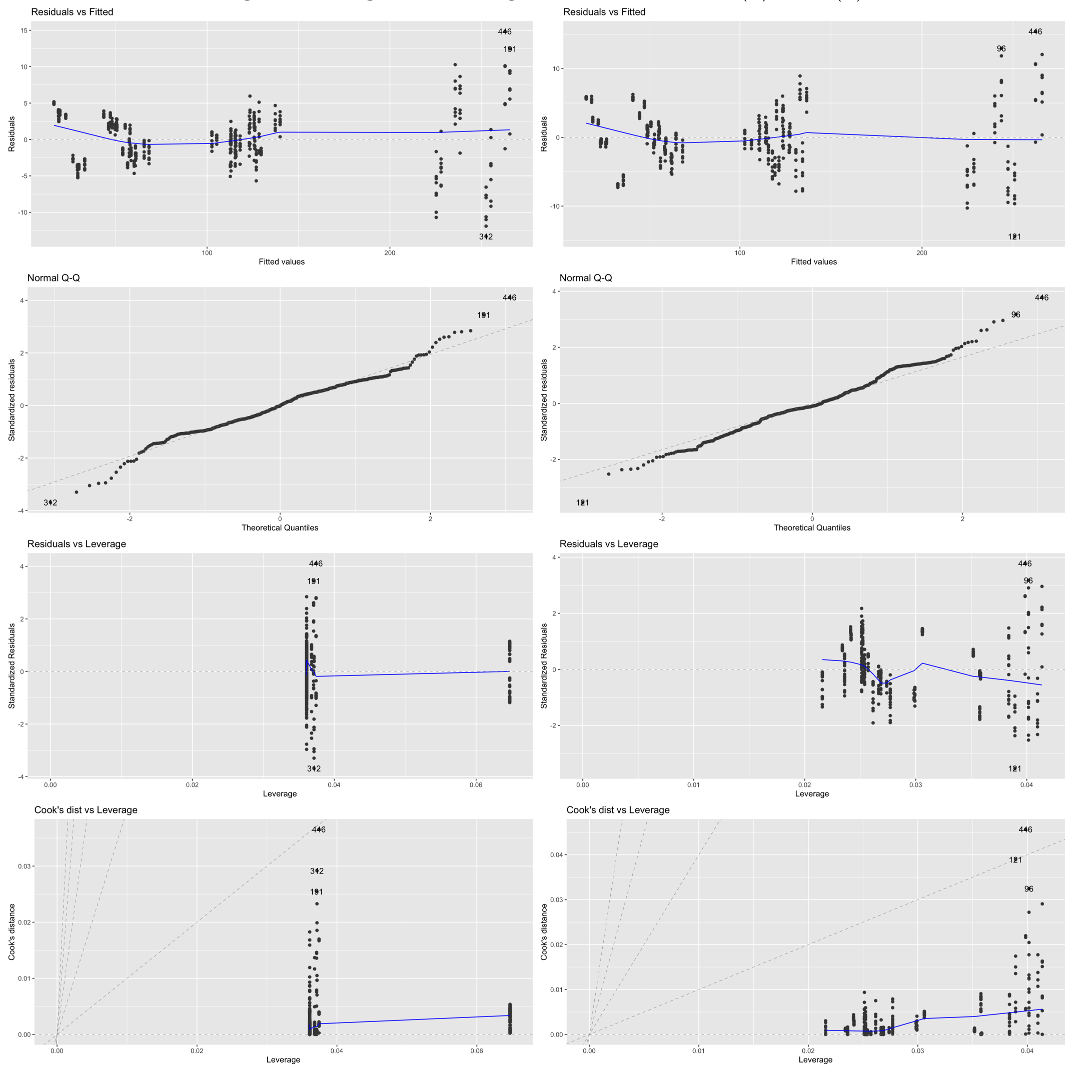
	Model 4	Model 6
	(1)	(2)
sexM	−2.981*** (0.930)	−2.568*** (0.352)
dist100:strokeBreaststroke		3.954*** (0.527)
dist200:strokeBreaststroke		10.976*** (0.749)
dist400:strokeBreaststroke		
dist100:strokeButterfly		−0.278 (0.582)
dist200:strokeButterfly		1.148 (0.758)
dist400:strokeButterfly		
dist100:strokeFreestyle		−2.285*** (0.512)
dist200:strokeFreestyle		−7.813*** (0.660)
dist400:strokeFreestyle		−22.552*** (1.283)
dist100:strokeMedley		−0.943 (0.735)
dist200:strokeMedley		
dist400:strokeMedley		
lengths:sexM		−1.511*** (0.080)
dist100	28.966*** (0.714)	30.528*** (0.413)
dist200	97.097*** (0.796)	97.174*** (0.645)
dist400	228.734*** (1.632)	238.544*** (1.431)
courseShort	−3.631*** (0.418)	
strokeBreaststroke	9.185*** (0.551)	3.214*** (0.442)
strokeButterfly	−0.398* (0.239)	−0.771 (0.515)
strokeFreestyle	−8.210*** (0.443)	−2.241*** (0.419)
strokeMedley	6.915*** (0.683)	2.697*** (0.603)
sexM:dist100	−3.300*** (1.040)	
sexM:dist200	−9.960*** (1.091)	
sexM:dist400	−19.395*** (2.090)	
lengths		0.091 (0.095)
Constant	30.356*** (0.689)	27.321*** (0.438)
Observations	446	446
R^2	0.997	0.999
Adjusted R^2	0.996	0.999
Akaike Inf. Crit.	2,552.413	1,983.252
Residual Std. Error	4.162 (df = 433)	2.184 (df = 427)
F Statistic	10,473.610*** (df = 12; 433)	25,406.290*** (df = 18; 427)
Note:		*p<0.1; **p<0.05; ***p<0.01

Before interpreting the estimation results, we should investigate if the model has high-leverage or high-influence points. Figure 4 below displays diagnostic plots for models (1) and (4). Carefully looking at the axes in the lower 4 plots, we can see that model 4 has fewer points with high leverage and residuals, indicating potentially fewer influential outliers. As the bottom two plots show, it also features fewer points with high Cook's distance, suggesting better robustness. In terms of homoscedasticity, Model (4) also performs better, as shown by the smaller range on the Y axis of the top two plots. Lastly, the residuals of model (4) are marginally closer to a normal distribution, as evidenced by the Q-Q plots in the second row. I conclude that model 4 improves on the baseline and will therefore use model 4 as my final model. There are no outliers of concerning influence or leverage. The last observation has a cook's distance value of about 0.045, which is above the rule-of-thumb threshold of $\frac{8}{n-2p} = \frac{8}{446-(2*13)} \approx 0.019$. Looking at the data, we can see that this point is not implausible:

```
tail(swim)
    dist stroke sex course   time lengths
441  400 Medley   F  Short 267.74      16
442  400 Medley   F  Short 267.78      16
443  400 Medley   F  Short 267.86      16
444  400 Medley   F  Short 272.98      16
445  400 Medley   F  Short 273.07      16
446  400 Medley   F  Short 277.79      16
```

I therefore conclude that there is no valid reason to drop it and will proceed to interpreting the model.

Figure 4: Regression Diagnostics for Models (1) and (4)



4 Interpretation

Table 4 below displays estimation results. As the diagnostic plots in Figure 4 do show some signs of heteroscedasticity, I use Huber-White robust standard errors. Subsequently mentioned coefficients are all estimated coefficients, not population-level ones. The intercept term, estimated at 30.35 with a very low p-value, represents the average time for the baseline group, female swimmers, in a long course, swimming a distance of 50 meters, in backstroke. The coefficients for 'sexM', 'dist100', 'dist200', and 'dist400' are all statistically significant, indicating that being male and increasing the distance of the swim both have substantial effects on time. Specifically, being male reduces the time by 2.98 seconds on average (all else being equal), which is enhanced further by interaction terms with distance ('sexM:dist100', 'sexM:dist200', 'sexM:dist400'), showing that the effect of sex on time varies with the distance swum. The interaction terms suggest that the reduction in time for males increases with the distance of the swim, indicating a possible advantage in longer distances.

The 'courseShort' coefficient is negative and significant, suggesting that swimming in a short course leads to a reduction in time by 3.63 seconds compared to a long course. The effects of stroke type are also significant: 'strokeBreaststroke' and 'strokeMedley' are associated with an increase in time by 9.1855 and 6.9146 seconds, respectively, while 'strokeFreestyle' leads to a decrease in time by 8.2098 seconds. 'strokeButterfly' does not have a significant effect on time. The model has a very high R-squared value (0.9966), which indicates that the model explains nearly all of the variability in swimming times. However, caution should be exercised as such a high R-squared value may also indicate overfitting. The residual standard error is 4.162 on 433 degrees of freedom, reflecting the average deviation of the observed times from the fitted values. The F-statistic is significantly large, indicating that the model as a whole is statistically significant.

Table 6: Regression Results

	Model 4
sexM	-2.981^{***} (0.930)
dist100	28.966^{***} (0.714)
dist200	97.097^{***} (0.796)
dist400	228.734^{***} (1.632)
courseShort	-3.631^{***} (0.418)
strokeBreaststroke	9.185^{***} (0.551)
strokeButterfly	-0.398^{*} (0.239)
strokeFreestyle	-8.210^{***} (0.443)
strokeMedley	6.915^{***} (0.683)
sexM:dist100	-3.300^{***} (1.040)
sexM:dist200	-9.960^{***} (1.091)
sexM:dist400	-19.395^{***} (2.090)
Constant	30.356^{***} (0.689)
Observations	446
R^2	0.997
Adjusted R^2	0.996
Akaike Inf. Crit.	2,552.413
Residual Std. Error	4.162 (df = 433)
F Statistic	$10{,}473.610^{***}$ (df = 12; 433)
Note:	$^{*}p<0.1$; $^{**}p<0.05$; $^{***}p<0.01$

5 Prediction

Using model (4), I obtain the following predicted times:

Table 7: Prediction Intervals

Prediction	Lower Bound	Upper Bound
250.88	242.54	259.22
30.36	22.00	38.71
258.69	250.29	267.09
66.24	57.92	74.55

The predictions from the linear model mod4 for the four additional races suggest the following:

- `RaceA` (400 Freestyle): Predicted time is 250.88 seconds with a prediction interval of 242.54 to 259.22 seconds, indicating a fairly consistent performance expected for a race of this length.

- `RaceB` (50 Backstroke): A short race with a predicted time of 30.36 seconds and a wider interval (22.00 to 38.71 seconds) reflecting higher uncertainty in such a brief event where starts and turns are crucial.

- `RaceC` (400 Butterfly): Similar predicted time as the 400 Freestyle at 258.69 seconds, with a prediction interval of 250.29 to 267.09 seconds, showing comparable expected performance across both 400m events.

- `RaceD` (100 Medley): Predicted time of 66.24 seconds, with an interval of 57.92 to 74.55 seconds, wider due to the complexity of the medley involving multiple strokes.

The intervals provide a realistic range for actual times, taking into account the model's input factors. These intervals are especially broad for the shorter distances, indicative of the inherent performance variability in sprint events.

6 Appendix: Source Code

```r
library(tidyverse)
library(ggplot2)
library(gridExtra)
library(stargazer)
library(kableExtra)
library(MASS)




## Set Paths for tables and figures
root = "/Users/ts/Git/Practicals"
tab = "/Users/ts/Library/CloudStorage/Dropbox/Apps/Overleaf/Practical Template/Tables"
fig = "/Users/ts/Library/CloudStorage/Dropbox/Apps/Overleaf/Practical Template/Figures"
code = "/Users/ts/Library/CloudStorage/Dropbox/Apps/Overleaf/Practical Template/Code"

if (getwd() != root) {
  setwd(root)
}
# Load data
data <- read.csv("swim.csv")
## Codebook ##
# For each event, the times of the finalists are recorded as well as some other
# information about the event. The variables recorded are:
# • event, the name of the event, e.g. “50 m Freestyle”
# • dist, the length of the event, in metres
# • stroke, the stroke swum in the event
# • sex, to indicate whether an event is women’s or men’s
# • course, to indicate whether an event is short course or long course
# • time, the time of one of the swimmers in the final, in seconds.

# Exercise:
#
# You are asked to investigate how race times depend on the other variables.
#
# The primary aim of the analysis is: (i) to obtain an interpretable model that
# explains how time depends on the other variables; and (ii) to interpret the
# model you obtain.
# You are also asked, using the same model, to predict times for four additional races.

# 1. Perform an exploratory analysis of the data and summarise your findings.
# You may wish to consider some numerical summaries as well as some exploratory plots.

head(data)
str(data)
```

```r
# Sex coded as string, needs to be factor
data$sex <- as.factor(data$sex)

unique(data$course)
# same thing for course
data$course <- as.factor(data$course)

unique(data$stroke)
# repeat for stroke
data$stroke <- as.factor(data$stroke)
str(data)

#repeat for event
data$event <- as.factor(data$event)

# event is concat of dist and stroke, has
length(unique(data$event)) # 16
# distinct levels, which blows up df used and ruins sensible interpretability
subset(data, dist == 400 & stroke == "Butterfly")
# we also have no current observations for 400m fly, which means we cannot use
# it to predict anyways

mod0a <- lm(time ~ dist + sex + course + stroke, data = data)
mod0b <- lm(time ~ . , data = data)
mod0c <- lm(time ~ dist*stroke + sex + course, data = data)
stargazer(mod0a, mod0b, mod0c, type = "text",
          single.row = T,
          omit = ".*",
          column.labels = c("Without 'event'", "With 'event'",
                            "with stroke * dist"))

# dropping event column makes a
# difference in terms of Adj R2, Residual SE

# this is likely because the event variable has a base level of 50 free, which
# is captured by the intercept. In the third model, the intercept captures the
# product of the base levels of both dist and stroke
# mod0b is more flexible, but also more prone to overfitting
# constrains the relationship by assuming that the effect of a particular
# combination is the sum of the main effects plus an interaction effect.
# This is less flexible but might be more stable if there are many combinations.
# Indeed we can see that there are
length(unique(data$dist)) * length(unique(data$stroke)) # 20 possible combinations
# of dist and stroke, but there are only
```

```r
length(levels(data$event)) # 16 observed combinations

# When we use event as a variable, Only the combinations that exist in the data
# will have their own coefficient. This means there will be 15 coefficients
# for the event variable (excluding the reference level).

# When we instead interact dist and stroke, he model attempts to estimate
# coefficients for all possible interactions, even if certain combinations don't
# exist in the data. This can result in unstable or non-sensical coefficients for
# the missing combinations. The behavior of lm in R, in this case, depends on the
# contrasts used for factors. By default, R uses treatment contrasts, which
# compare each level to a reference level. If certain interactions don't exist,
# they won't get their own coefficient, but they can still influence the
# coefficients of the main effects.

# In conclusion, since we can't use the event variable in prediction later on,
# I drop it here, as optimizing models will lead to inclusion of this variable
# which we cannot use to predict

# New Object for clean data
swim = data[, -1] # drop event for reasons stated above

# Dist is technically numerical but practically has
length(unique(swim$dist)) # 4 distinct levels
# A classical ceteris paribus interpretation also does not really make sense
# as race distances are fixed and "increasing by 1 unit" does not have a sensible
# interpretation

# Check whether it makes a difference for linear modelling
swim$dist <- as.factor(swim$dist)
mod0d <- lm(time ~ dist*stroke + sex + course , data = data[, -1])
mod0e <- lm(time ~ dist*stroke + sex + course , data = swim)
stargazer(mod0d, mod0e, type = "text",
          single.row = T,
          omit = ".*",
          title = "Full Model with 'dist' cast as",
          column.labels = c("Numeric", "Factor"))

# it does indeed make a (small) difference, factor takes up more df, but negligible

# Summary Table
swim %>%
  group_by(sex, course, stroke) %>%
  summarize(
    Avg_Time = mean(time, na.rm = TRUE),
    SD_Time = sd(time, na.rm = TRUE),
    .groups = 'drop'
  )

stargazer(swim, type = "text")

hist <- ggplot(swim, aes(x = time)) +
  geom_histogram(binwidth = 1) +
  facet_wrap(~stroke)

box <- ggplot(swim, aes(x = stroke, y = time, fill = sex)) +
  geom_boxplot() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  facet_wrap(~course)

point <- ggplot(swim, aes(x = stroke, y = time, color = sex, shape = factor(dist))) +
  geom_point(position = position_dodge(0.8)) +
  facet_wrap(~course) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(x = "Stroke", y = "Time (seconds)")

violin1 <- ggplot(swim, aes(x = stroke, y = time, fill = sex)) +
  geom_violin(position = position_dodge(0.8), width = 0.7) +
  theme_minimal() +
  labs(x = "Stroke", y = "Time (seconds)") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_fill_manual(values = c("F" = "pink", "M" = "lightblue"))


violin2 <- ggplot(swim, aes(x = stroke, y = time, fill = course)) +
  geom_violin(position = position_dodge(0.8), width = 0.7) +
  theme_minimal() +
  labs(x = "Stroke", y = "Time (seconds)") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_fill_manual(values = c("Long" = "lightgreen", "Short" = "cyan"))

violins = grid.arrange(violin1, violin2, ncol=2)

violin3 <- ggplot(swim, aes(x = sex, y = time, fill = course)) +
  geom_violin(position = position_dodge(0.8), width = 0.7) +
  theme_minimal() +
  labs(x = "Stroke", y = "Time (seconds)") +
  scale_fill_manual(values = c("Long" = "lightgreen", "Short" = "cyan"))
```

```r
# Collect in list
plotlist <- list(hist, box, point, violins, violin3)

## 2 Model building:

# establish baseline
mod1 <- lm(time ~ dist*stroke + sex + course , data = swim)
mod2 <- lm(time ~ dist + stroke + sex + course , data = swim)

stargazer(mod1, mod2, type = "text",
          single.row = T,
          omit = ".*",
          title = "Baseline: 'dist' and 'factor' included as",
          column.labels = c("Interaction", "Additive"))
# Interaction better across the board



# Diagnostic plots for mod1
par(mfrow = c(2, 2))
plot(mod1)
# RvF shows slight pattern
# Q-Q shows deviance in tails
# S-L suggests hetsked

# AIC
step_aic <- stepAIC(mod1, direction = "backward")
# AIC deteriorates when we drop predictors

# BIC
step_bic <- step(mod1, direction = "backward", k = log(nrow(swim)))
# BIC also deteriorates when we drop predictors

# Interaction terms
# maximal interactions:
mod3 <- lm(time ~ dist*course + stroke*sex, data = swim)

# based on plot "point", Different strokes have varying times across both short
# and long distances.The time differences between strokes in short vs.
# long distances are not consistent. For instance, the time gap for Butterfly
# between short and long distances is slightly more pronounced than for Backstroke.
mod4 <- lm(time ~ stroke*dist + sex + course, data = swim)

# Males and females have different times for the same strokes. For example, in
# Freestyle and Medley (both short and long), females tend to have slightly
# longer times than males.
```

```
mod5 <- lm(time ~ sex*stroke + dist + course, data = swim)

# For certain strokes, the time differences between males and females appear
# more pronounced in long distances compared to short distances.
mod6 <- lm(time ~ sex*dist + course + stroke, data = swim)

# create new variable that captures lengths swum
swim$lengths <- ifelse(swim$course == "Short",
                       as.numeric(as.character(swim$dist)) / 25,
                       as.numeric(as.character(swim$dist)) / 50)

mod7 <- lm(time ~ sex * stroke + lengths, data = swim)
mod8 <- lm(time ~ dist * stroke + lengths*sex, data = swim)

# get all models into list
model_list <- mget(grep("^mod[1-9]$", ls(), value = TRUE))

#compare ICs:
aic <- round(sapply(model_list, AIC), 2)
bic <- round(sapply(model_list, BIC), 2)
which(aic ==min(aic))
which(bic ==min(bic))
# model 7 is best in terms of AIC and BIC
# get number of coefficients
p <- sapply(model_list, function(mod) length(coef(mod)))

# Make small table
icmat = rbind(p, aic, bic)
colnames(icmat) <- paste0("Model ", 1:8)
icmat[1, ] <- as.integer(icmat[1, ])
ictable <- kable(icmat, "latex", booktabs = TRUE, digits = 2) %>%
  kable_styling(latex_options = c("striped", "scale_down")) %>%
  row_spec(0, bold = TRUE) # Make the header row bold




# However, can't use event as factor because interacting it destroys df and
# interpretability


# Diagnostic plots for mod6
par(mfrow = c(2, 2))
plot(mod6)

# Diagnostic plots for mod5
```

```r
par(mfrow = c(2, 2))
plot(mod5)

# Data for prediction intervals
new_races <- data.frame(
  name = c("RaceA", "RaceB", "RaceC", "RaceD"),
  dist = c(400, 50, 400, 100),
  stroke = c("Freestyle", "Backstroke", "Butterfly", "Medley"),
  sex = rep("F", 4),
  course = rep("Long", 4)
)

# Apply data transformations
# Sex coded as string, needs to be factor
new_races$sex <- as.factor(new_races$sex)
# same thing for course
new_races$course <- as.factor(new_races$course)
# repeat for stroke
new_races$stroke <- as.factor(new_races$stroke)
# repeat for dist
new_races$dist <- as.factor(new_races$dist)

# New Object for clean data
swim_test = new_races

# Do Prediction
pred_df <- data.frame(Predictions = predict(mod6, newdata = swim_test))


## Export tables
setwd(tab)
save_kable(ictable, file = "ictable.tex")

kable(pred_df, "latex", booktabs = TRUE) %>%
  kable_styling(latex_options = c("striped", "scale_down")) %>%
  save_kable(file = "predictions.tex")

stargazer(mod0a, mod0b, mod0c, out = "event.tex",
          single.row = T,
          omit = ".*",
          title = "Inclusion of 'event' variable",
          column.labels = c("Without 'event'", "With 'event'",
                            "with stroke * dist"))

stargazer(mod0d, mod0e, out = "dist.tex",
          single.row = T,
          omit = ".*",
          title = "Full Model with 'dist' cast as",
          column.labels = c("Numeric", "Factor"))

stargazer(mod1, mod2, out = "dist_stroke.tex",
          single.row = T,
          omit = ".*",
          title = "Baseline: 'dist' and 'stroke' included as",
          column.labels = c("Interaction", "Additive"))

# Prep reg table with robust SEs for better inference given Heterosked.
library(sandwich)
robustSE6 = vcovHC(mod6)
robustSE1 = vcovHC(mod1)

stargazer(mod1, mod6, title="Regression Results", label="tab:results",
          se=list(sqrt(diag(robustSE1)), sqrt(diag(robustSE6))),
          type="latex",
          align=TRUE,
          out = "regtable.tex",
          column.labels=c("Initial Model", "Final Model"),
          dep.var.caption="",
          dep.var.labels.include = FALSE,
          no.space=TRUE,
          single.row=TRUE,
          header=FALSE,
          digits=3)
# omit.stat=c("adj.rsq"), #, "f", "ser"
setwd(root)


## Export figures
# Side-by-side of baseline and mod7
# Set up plotting parameters
setwd(fig)
png("diag_s_by_s.png", width = 1400, height = 2800, res = 80)

# Adjust layout and margins
par(mfrow = c(8, 2), mar = c(2, 4, 2, 2) + 0.1, oma = c(0, 1, 0, 0), cex = 0.8)

# Plot diagnostics for mod1 and mod6 side by side
plot(mod1, which = 1) # main = "Model 1: Residuals vs Fitted")
plot(mod6, which = 1) #, main = "Model 7: Residuals vs Fitted")
plot(mod1, which = 2) #, main = "Model 1: Normal Q-Q")
```

```r
plot(mod6, which = 2) #, main = "Model 7: Normal Q-Q")
plot(mod1, which = 3) #, main = "Model 1: Scale-Location")
plot(mod6, which = 3) #, main = "Model 7: Scale-Location")
plot(mod1, which = 5) #, main = "Model 1: Residuals vs Leverage")
plot(mod6, which = 5) #, main = "Model 7: Residuals vs Leverage")
dev.off()

# export ggplots
# Loop through the list of plots
for (i in seq_along(plotlist)) {
  if (!is.null(plotlist[[i]])) { # Check if the plot is not null
    # Construct the filename using the index
    filename <- paste0("plot", i, ".png")
    # Save the plot to disk
    ggsave(plot = plotlist[[i]], filename = filename)
  }
}



setwd(code)
file.copy(from = "/Users/ts/Git/Practicals/Linear-Models-marked-practical.R",
          to = "/Users/ts/Library/CloudStorage/Dropbox/Apps/Overleaf/Practical Template/Code")
```