МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«Дальневосточный федеральный университет»
(ДВФУ)

ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ
Департамент информационных и компьютерных систем

Курс «Компьютерные методы анализа больших данных»

Лабораторная работа №2
Контрольное мероприятие по рейтингу

на тему «Очистка и сравнительный анализ интернет-данных»
вариант №14

Выполнил студент Б9122-01.03.02мкт
Пелагеев Д.И.
_____

Проверил доцент Достовалов В.Н.

г. Владивосток
2025

Оглавление

Введение

В рамках данной работы проводится анализ данных опроса студентов курса «Stat 100» Университета Иллинойса за осенний семестр 2018 года. Основная цель анализа — изучить взаимосвязь между количеством напитков, потребляемых в неделю (drinks_per_week), временем, проводимым на вечеринках (party_hours_per_week), полом (gender) и типом населённого пункта (home_bc), в котором проживает студент.

В ходе выполнения работы были реализованы следующие этапы:

а) Загрузка и предварительная обработка исходного файла данных Stat 100 Survey 2, Fall 2018.dat, включая очистку и структурирование информации.

б) Преобразование и кодирование категориальных переменных, в том числе модификация столбца Home town для формирования базового уровня.

в) Построение двух моделей линейной регрессии:

— Модель m1: зависимость drinks_per_week от party_hours_per_week и gender;

— Модель m2: зависимость drinks_per_week от party_hours_per_week, gender и home_bc.

г) Визуализация результатов обеих регрессий на одном графике для сравнения.

д) Проведение прогнозирования по модели m1 с построением доверительных и предиктивных интервалов.

Анализ подобных данных позволяет выявить, какие социально-демографические и поведенческие факторы оказывают влияние на уровень потребления алкогольных напитков среди студентов, что важно для последующего построения профилактических и образовательных программ.

1 Подготовка данных

Исходный файл `Stat 100 Survey 2, Fall 2018.dat` содержит анкетные данные студентов. Для начала были считаны исходные заголовки, которые содержали лишние пояснения и пробелы. После первичной очистки (удаления скобок и лишних пробелов) они выглядели так:

Исходные имена признаков после очистки:

```
raw_header   <- readLines(DATA_FILE, n = 1)
clean_header <- gsub("\\s*\\([^()]*\\)", "", raw_header)
clean_header <- gsub("\\s+", " ", clean_header)
clean_header <- trimws(clean_header)


header_tokens <- strsplit(clean_header, " ")[[1]]
col_names_raw <- header_tokens[-1]
col_names_raw <- col_names_raw[-2]
col_names_raw
```

Первично обработанный список содержит 32 переменные, включая как числовые, так и категориальные.

```
"Gender", "Greek", "Home_Town", "In_State", "Ethnicity",
"Religion", "Religious","like_Math", "Calculus", "ACT",
"GPA", "Party_Hours_per_week", "Drinks_per_week",
"Sex_Partners", "Relationships", "Call_Parents",
"Cell_Phone", "Social_Media", "Texts", "Good_or_Well",
"Expected_Income", "President", "Illinois_Commitment",
"Family_Income", "Liberal"  "Political_Party", "Grade_vs_Learning",
"Parent_Relationship", "Work_Hours", "Tuition", "Career", "Section"
```

Преобразование имён к стилю snake_case:

Для удобства анализа имена признаков были приведены к единому стилю, без пробелов и с нижним подчёркиванием:

```
col_names <- make_snake(col_names_raw)
```

Результат:

```
"gender", "greek", "home_town", "in_state", "ethnicity",
"religion", "religious", "like_math", "calculus", "act", "gpa",
```

"party_hours_per_week", "drinks_per_week", "sex_partners", "relationships", "call_parents", "cell_phone", "social_media", "texts", "good_or_well", "expected_income", "president", "illinois_commitment", "family_income", "liberal", "political_party", "grade_vs_learning", "parent_relationship", "work_hours", "tuition", "career", "section"

Пошаговое формирование таблицы:

Сначала данные были считаны из файла. Первая строка с названиями была пропущена, а лишние технические столбцы удалены:

```
df <- read.table(DATA_FILE, header = FALSE, sep = "", stringsAsFactors =
    FALSE, skip = 1, fill = TRUE)
head(df)
```

Таблица 1 — Фрагмент таблицы после чтения файла (ещё без имён признаков)

V1	V2	V3	...	V32	V33	V34	V35
Obs1	1	1	...	10	40	5	1
Obs2	2	1	...	20	100	3	1
Obs3	3	1	...	0	100	7	1
...	.	.	...	...	...	...	.

Затем удалялись две первые технические колонки (номера и дублирующий столбец):

```
df <- df[, -c(1:2)]
df <- df[, -2]
head(df)
```

Таблица 2 — Фрагмент таблицы после удаления служебных столбцов

V3	V5	V6	...	V32	V33	V34	V35
1	0	3	...	10	40	5	1
1	0	2	...	20	100	3	1
1	1	2	...	0	100	7	1
...	.	.	...	...	...	...	.

Далее таблице были присвоены очищенные и стандартизированные имена столбцов:

```
names(df) <- col_names
head(df)
```

Таблица 3 — Фрагмент таблицы с корректными названиями признаков

gender	greek	home_town	...	family_income	career	section
1	0	3	...	85	5	1
1	0	2	...	80	3	1
1	1	2	...	50	7	1
...	.	.	...	...	.	.

Следующим шагом была обработка и перекодировка категориальных переменных. Были выделены основные факторы и преобразованы в `factor`:

```
factor_cols <- c("gender", "greek", "home_town", "in_state", "ethnicity"
    ,
                 "religion", "calculus", "cell_phone", "president",
                 "political_party", "section")
factor_cols <- intersect(factor_cols, names(df))
df[factor_cols] <- lapply(df[factor_cols], factor)
```

Проверка уровней факторных переменных дала следующий результат:

```
$gender: [1] "0" "1"
$home_town: [1] "0" "1" "2" "3"
$cell_phone: [1] "0" "1" "2"
 и т.д.
```

После этого проведена фильтрация: из всех числовых столбцов были удалены строки с некорректными (отрицательными или равными 999) значениями. Затем была проверена и подтверждена полная отсутствия пропущенных значений:

```
numeric_mask <- sapply(df, is.numeric)
problem_mat  <- df[, numeric_mask, drop = FALSE] < 0 | df[, numeric_mask
    , drop = FALSE] == 999
```

```
rows_bad      <- rowSums(problem_mat, na.rm = TRUE) > 0
df <- df[!rows_bad, ]
colSums(is.na(df))
```

В завершение был введён дополнительный фактор — бинарная переменная `home_bc`, где "big_city"выделен как отдельная группа, а остальные категории объединены:

```
df$home_bc <- factor(ifelse(df$home_town == "3", 1, 0),
                     levels = c(0, 1), labels = c("other", "big_city"))
```

Для построения моделей рабочий датафрейм был сокращён до ключевых переменных:

```
df_work <- df[, c("gender", "party_hours_per_week", "drinks_per_week", "
    home_bc")]
head(df_work)
```

Таблица 4 — Финальная таблица для моделирования (фрагмент)

gender	party_hours_per_week	drinks_per_week	home_bc
1	4	12	big_city
1	3	1	other
1	10	50	other
1	1	1	other
1	0	0	big_city
0	4	3	other

Таким образом, процесс подготовки данных включает в себя пошаговое преобразование структуры, удаление некорректных и неинформативных наблюдений, а также перекодировку категориальных признаков — всё это позволило получить чистый, структурированный датасет, полностью готовый к последующему анализу и моделированию.

2 Построение моделей и визуализация

На следующем этапе были построены две линейные регрессионные модели, в которых целевой переменной выступает количество напитков, потребляемых в неделю (`drinks_per_week`), а объясняющими — количество часов, проводимых на вечеринках (`party_hours_per_week`), пол (`gender`), а также тип населённого пункта (`home_bc`).

Для первой модели (**m1**) использовались переменные `party_hours_per_week` и `gender`:

```
m1 <- lm(drinks_per_week ~ party_hours_per_week + gender, data = df_work
    )
```

Вторая модель (**m2**) дополнительно учитывает переменную `home_bc` (тип населённого пункта):

```
m2 <- lm(drinks_per_week ~ party_hours_per_week + gender + home_bc, data
    = df_work)
```

В обеих моделях пол и тип населённого пункта выступают категориальными предикторами.

Для наглядной интерпретации были построены графики, на которых визуализированы зависимости для обеих моделей.

Визуализация модели m1

На графике 1 представлены результаты регрессии m1. По оси X отложено число часов на вечеринках в неделю, по оси Y — количество потребляемых напитков в неделю. Для каждой группы по полу отображены свои линии регрессии, а также области доверительных и предиктивных интервалов (полупрозрачные полосы).

Доверительный интервал показывает, где находится истинная средняя линия регрессии для каждого пола с заданной вероятностью (обычно 95%). Предиктивный интервал шире и показывает, где с большой вероятностью окажется индивидуальное новое наблюдение.

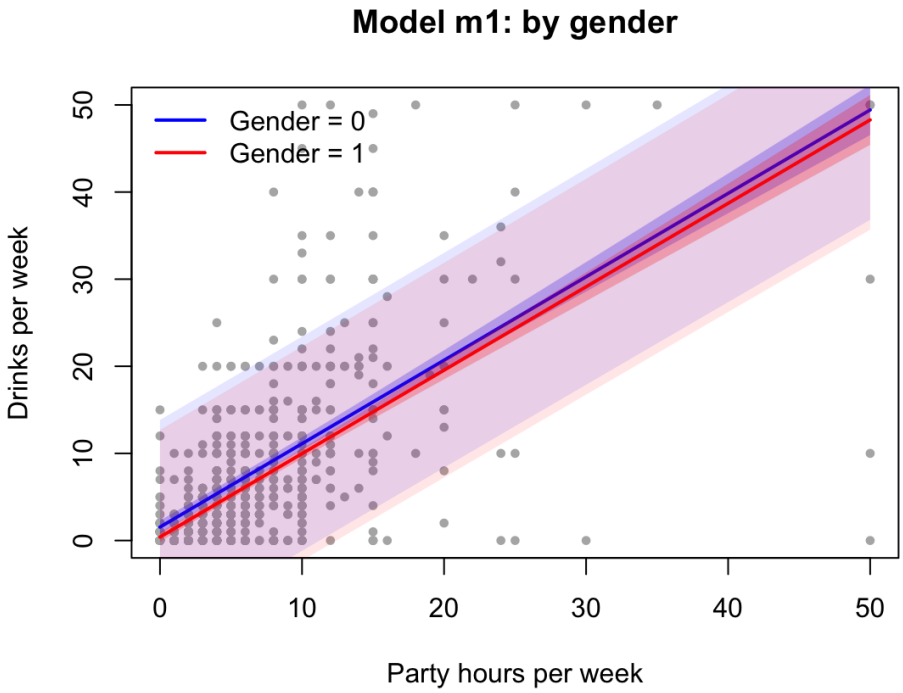

Рисунок 1 — Регрессия m1: зависимость drinks_per_week от party_hours_per_week и gender. Доверительные (шире) и предиктивные интервалы для каждого пола.

Визуализация модели m2

На втором графике представлены результаты второй модели, которая учитывает и пол, и тип населённого пункта. На одной оси X — часы на вечеринках, на оси Y — напитки в неделю. Для каждой из четырёх комбинаций "пол × home_bc"проведена отдельная линия.

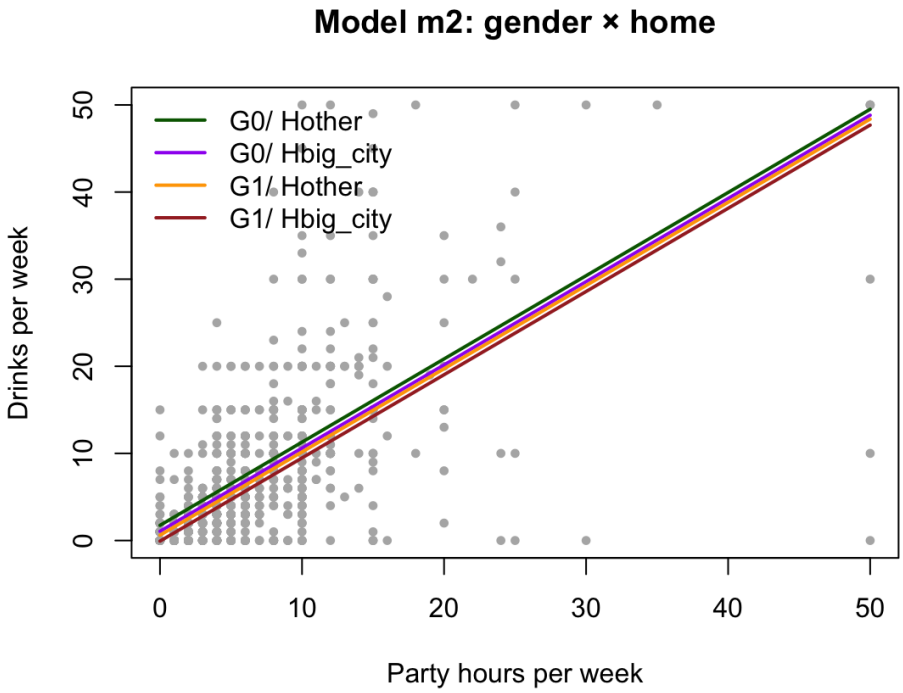

Рисунок 2 — Регрессия m2: зависимость drinks_per_week от party_hours_per_week, gender и home_bc.

Как видно из графиков, во всех моделях основной вклад вносит переменная `party_hours_per_week`, а разница между группами по полу и типу города визуально минимальна — линии расположены очень близко друг к другу.

Коэффициенты моделей

В результате оценки первой модели (m1), в которую входят переменные party_hours_per_week и gender, получены следующие значения коэффициентов:

Таблица 5 — Коэффициенты линейной модели m1

Переменная	Оценка	Std. Error	t value	p-value
(Intercept)	1.54	0.39	3.97	< 0.001
party_hours_per_week	0.96	0.03	29.99	< 0.001
gender1	-1.14	0.42	-2.74	0.006

Добавление в модель m2 переменной home_bc практически не изменяет оценки других коэффициентов. Эффект принадлежности к группе big_city оказывается статистически незначимым:

Таблица 6 — Коэффициенты линейной модели m2

Переменная	Оценка	Std. Error	t value	p-value
(Intercept)	1.74	0.41	4.27	< 0.001
party_hours_per_week	0.96	0.03	29.91	< 0.001
gender1	-1.13	0.42	-2.71	0.007
home_bcbig_city	-0.68	0.43	-1.60	0.109

Во всех моделях наибольший вклад вносит переменная party_hours_per_week: увеличение на 1 час связано практически с одним дополнительным напитком в неделю. Эффект пола также значим — при gender = 1 количество напитков в среднем меньше на 1.13–1.14, чем при gender = 0. Переменная home_bc (big city) статистически незначима: её коэффициент отрицательный, но уровень значимости превышает 0.05. Доля объяснённой дисперсии (R^2) для обеих моделей составляет около 46%, что подтверждает — основная тенденция хорошо описывается, однако остаётся значительный индивидуальный разброс в данных.

3 Прогнозирование по модели m1 (доверительный и предиктивный интервалы)

Для анализа практического применения модели m1 было выполнено построение доверительных и предиктивных интервалов для ожидаемого значения и для отдельных новых наблюдений соответственно. Это позволяет оценить, насколько устойчивы прогнозы и каков возможный разброс индивидуальных значений.

Доверительный интервал (confidence interval) показывает диапазон, в котором с заданной вероятностью находится истинное среднее значение отклика для заданных значений факторов. Предиктивный интервал (prediction interval) шире и показывает диапазон, в котором с этой же вероятностью окажется значение отклика для нового индивидуального наблюдения с заданными характеристиками.

Вычисление и визуализация интервалов выполнялись для обеих групп по полу, на всём диапазоне переменной `party_hours_per_week`. В качестве нового "X"бралась равномерная сетка значений, покрывающая весь диапазон наблюдений.

Пример кода для вычисления интервалов:

```
xseq <- seq(min(df_work$party_hours_per_week, na.rm = TRUE),
            max(df_work$party_hours_per_week, na.rm = TRUE), length.out
               = 120)

# Для группы gender = 0:
nd1 <- data.frame(party_hours_per_week = xseq,
                  gender = factor(rep(levels(df_work$gender)[1], length(
                     xseq)), levels = levels(df_work$gender)))
pred_ci1 <- predict(m1, newdata = nd1, interval = "confidence")
pred_pi1 <- predict(m1, newdata = nd1, interval = "prediction")

# Для gender = 1 аналогично.
```

На графике (см. рис. 1) эти интервалы представлены полупрозрачными областями: доверительный интервал (уже), предиктивный (шире), отдельно для каждой группы по полу. Центральная линия — это прогноз модели для среднего значения отклика.

Практическое значение:

— Доверительный интервал используется для оценки того, насколько точно мы можем предсказать среднее значение потребления алкоголя среди студентов с определёнными характеристиками.

— Предиктивный интервал показывает, какой разброс можно ожидать для индивидуального студента при тех же характеристиках (например, если взять случайного студента с заданным числом часов вечеринок и полом).

В данном случае оба интервала довольно узкие при средних значениях и расширяются на краях диапазона, что типично для линейной регрессии. Разница между полами (gender) остаётся минимальной во всём диапазоне.

Заключение

В ходе данной работы была реализована последовательная обработка и анализ данных опроса студентов Университета Иллинойса с целью изучить взаимосвязь между потреблением алкогольных напитков, временем, проводимым на вечеринках, полом и типом населённого пункта проживания.

В соответствии с поставленными целями:

— Проведена очистка и структурирование исходного файла данных, в том числе перекодировка и сокращение переменных, что позволило получить качественный и пригодный для анализа датафрейм без пропусков и аномалий.

— Построены две регрессионные модели: первая учитывала только число часов на вечеринках и пол, вторая — дополнительно тип населённого пункта.

— Выполнена визуализация результатов, что позволило наглядно сравнить влияние разных факторов.

— Проведено прогнозирование с использованием доверительных и предиктивных интервалов, что дало возможность оценить точность модели и возможный разброс индивидуальных значений.

Основные результаты анализа:

— Главным фактором, определяющим уровень потребления алкоголя среди студентов, является количество времени, проведённого на вечеринках. В среднем, каждый дополнительный час на вечеринке увеличивает ожидаемое количество напитков на один.

— Пол и тип населённого пункта оказывают минимальное влияние: различия между группами незначимы как статистически, так и визуально.

— Модель даёт достаточно узкие доверительные интервалы для среднего значения, однако предиктивные интервалы (для индивидуальных прогнозов) шире, что отражает высокую вариабельность индивидуального поведения студентов.

Таким образом, поставленные во введении цели были полностью достигнуты: была проанализирована роль социально-демографических и поведенческих факторов в формировании потребления алкоголя среди студентов, и подтверждено, что именно стиль досуга (а не пол или город) является основным предиктором. Результаты работы могут быть полезны для разработки профилактических и образовательных программ, направленных на снижение уровня потребления алкоголя среди студенческой молодёжи.

Список использованных источников

1. Официальная страница среды статистического моделирования R. — 2024. — Режим доступа: R: The R Project for Statistical Computing (дата обращения: 21.01.2025).

2. Официальная страница интегрированной среды разработки RStudio. — 2024. — Режим доступа: RStudio | Open source & professional software for data science teams - RStudio (дата обращения: 21.01.2025).

3. *Wickham, Hadley.* ggplot2: Elegant Graphics for Data Analysis (Use R!) / Hadley Wickham. — New York: Springer, 2009. — Режим доступа: ggplot2 (дата обращения: 21.01.2025).

4. *Max Kuhn.* The caret Package. — 2019. — Режим доступа: The caret Package (дата обращения: 21.01.2025).

5. *Alexandros Karatzoglou.* Kernel-Based Machine Learning Lab. — 2024. — Режим доступа: Kernel-Based Machine Learning Lab (дата обращения: 21.01.2025).

Приложение А Полный код на R

```r
setwd("/Users/daniil/Desktop/Обучение/BigData")

DATA_FILE <- "lab2_data.dat"

raw_header   <- readLines(DATA_FILE, n = 1)
clean_header <- gsub("\\s*\\([^()]*\\)", "", raw_header)
clean_header <- gsub("\\s+", " ", clean_header)
clean_header <- trimws(clean_header)


header_tokens <- strsplit(clean_header, " ")[[1]]
col_names_raw <- header_tokens[-1]
col_names_raw <- col_names_raw[-2]
col_names_raw
length(col_names_raw)

make_snake <- function(x) {
  x <- tolower(gsub("[^A-Za-z0-9]+", "_", x))
  # ensure uniqueness (append _n where needed)
  dup <- duplicated(x)
  if (any(dup)) x[dup] <- paste0(x[dup], "_", seq_along(which(dup)))
  x
}
col_names <- make_snake(col_names_raw)
length(col_names)


df <- read.table(
  DATA_FILE,
  header           = FALSE,
  sep              = "",
  stringsAsFactors = FALSE,
  skip             = 1,
  comment.char     = "",
  fill             = TRUE
)

head(df)
df <- df[, -c(1:2)]
df <- df[, -2]
head(df)
dim(df)


names(df) <- col_names
```

```r
head(df)


factor_cols <- c(
  "gender", "greek", "home_town", "in_state", "ethnicity", "religion",
  "calculus", "cell_phone", "president", "political_party", "section"
)


factor_cols <- intersect(factor_cols, names(df))

df[factor_cols] <- lapply(df[factor_cols], factor)

numeric_mask <- sapply(df, is.numeric)
problem_mat  <- df[, numeric_mask, drop = FALSE] < 0 | df[, numeric_mask
    , drop = FALSE] == 999
rows_bad     <- rowSums(problem_mat, na.rm = TRUE) > 0

df <- df[!rows_bad, ]

cat("\n\n--- Проверка уровней факторов ---\n")
print(lapply(df[factor_cols], levels))

cat("\n--- Кол-во пропущенных значений ---\n")
print(colSums(is.na(df)))


df$home_bc <- factor(ifelse(df$home_town == "3", 1, 0),
                     levels = c(0, 1),
                     labels = c("other", "big_city"))
head(df)
df_work <- df[, -c(2:11)]
head(df_work)
df_work <- df_work[, -c(4:22)]
head(df_work)

m1 <- lm(drinks_per_week ~ party_hours_per_week + gender, data = df_work
    )
summary(m1)
m2 <- lm(drinks_per_week ~ party_hours_per_week + gender + home_bc, data
     = df_work)
summary(m2)



xseq <- seq(min(df_work$party_hours_per_week, na.rm = TRUE),
```

```
            max(df_work$party_hours_per_week, na.rm = TRUE), length.out
                = 120)


col_gender <- c("blue", "red")
names(col_gender) <- levels(df_work$gender)[1:2]

col_combo <- c("darkgreen", "purple", "orange", "brown")
combo_labels <- NULL


plot(df_work$party_hours_per_week, df_work$drinks_per_week,
     xlab = "Party hours per week", ylab = "Drinks per week",
     main = "Model m1: by gender", pch = 19, cex = 0.6, col = gray(0.7))

g1 <- levels(df_work$gender)[1]
g2 <- levels(df_work$gender)[2]


nd1 <- data.frame(party_hours_per_week = xseq,
                  gender = factor(rep(g1, length(xseq)), levels = levels
                      (df_work$gender)))
pred_ci1 <- predict(m1, newdata = nd1, interval = "confidence")
pred_pi1 <- predict(m1, newdata = nd1, interval = "prediction")
base_col1 <- col_gender[g1]
ci_col1   <- adjustcolor(base_col1, alpha.f = 0.25)
pi_col1   <- adjustcolor(base_col1, alpha.f = 0.10)

polygon(c(xseq, rev(xseq)), c(pred_pi1[, "lwr"], rev(pred_pi1[, "upr"]))
    , col = pi_col1, border = NA)
polygon(c(xseq, rev(xseq)), c(pred_ci1[, "lwr"], rev(pred_ci1[, "upr"]))
    , col = ci_col1, border = NA)
lines(xseq, pred_ci1[, "fit"], col = base_col1, lwd = 2)

nd2 <- data.frame(party_hours_per_week = xseq,
                  gender = factor(rep(g2, length(xseq)), levels = levels
                      (df_work$gender)))
pred_ci2 <- predict(m1, newdata = nd2, interval = "confidence")
pred_pi2 <- predict(m1, newdata = nd2, interval = "prediction")
base_col2 <- col_gender[g2]
ci_col2   <- adjustcolor(base_col2, alpha.f = 0.25)
pi_col2   <- adjustcolor(base_col2, alpha.f = 0.10)

polygon(c(xseq, rev(xseq)), c(pred_pi2[, "lwr"], rev(pred_pi2[, "upr"]))
    , col = pi_col2, border = NA)
```

```
polygon(c(xseq, rev(xseq)), c(pred_ci2[, "lwr"], rev(pred_ci2[, "upr"]))
    , col = ci_col2, border = NA)
lines(xseq, pred_ci2[, "fit"], col = base_col2, lwd = 2)

legend("topleft", legend = paste("Gender =", c(g1, g2)), lwd = 2, col =
    c(base_col1, base_col2), bty = "n")


plot(df_work$party_hours_per_week, df_work$drinks_per_week,
     xlab = "Party hours per week", ylab = "Drinks per week",
     main = "Model m2: gender    home", pch = 19, cex = 0.6, col = gray
         (0.7))


h1 <- levels(df_work$home_bc)[1]
h2 <- levels(df_work$home_bc)[2]
combo_cols <- c("darkgreen", "purple", "orange", "brown")

nd11 <- data.frame(party_hours_per_week = xseq,
                   gender = factor(rep(g1, length(xseq)), levels =
                       levels(df_work$gender)),
                   home_bc = factor(rep(h1, length(xseq)), levels =
                       levels(df_work$home_bc)))
lines(xseq, predict(m2, newdata = nd11), col = combo_cols[1], lwd = 2)


nd12 <- nd11
nd12$home_bc <- factor(rep(h2, length(xseq)), levels = levels(
    df_work$home_bc))
lines(xseq, predict(m2, newdata = nd12), col = combo_cols[2], lwd = 2)


nd21 <- nd11
nd21$gender <- factor(rep(g2, length(xseq)), levels = levels(
    df_work$gender))
nd21$home_bc <- factor(rep(h1, length(xseq)), levels = levels(
    df_work$home_bc))
lines(xseq, predict(m2, newdata = nd21), col = combo_cols[3], lwd = 2)


nd22 <- nd21
nd22$home_bc <- factor(rep(h2, length(xseq)), levels = levels(
    df_work$home_bc))
lines(xseq, predict(m2, newdata = nd22), col = combo_cols[4], lwd = 2)

legend("topleft",
```

```
      legend = c(
        paste("G", g1, "/ H", h1, sep = ""),
        paste("G", g1, "/ H", h2, sep = ""),
        paste("G", g2, "/ H", h1, sep = ""),
        paste("G", g2, "/ H", h2, sep = "")
      ),
      lwd = 2, col = combo_cols, bty = "n")
```