统计机器学习 Homework 02

姓名: 雍崔扬
学号: 21307140051

我们定义多元线性回归的条件正态模型 $y \sim N(X\beta, \sigma^2 I_n)$
其等价形式为: $y = X\beta + \varepsilon$ 且 $\varepsilon \sim N(0_n, \sigma^2 I_n)$
其中 $\beta \in \mathbb{R}^{p+1}$ 为参数向量，$(x_1, y_1), \dots, (x_n, y_n)$ 为给定的样本数据，$x_i \in \{1\} \times \mathbb{R}^p$
我们假定设计矩阵 $X = [x_1, \dots, x_n]^{\mathrm{T}} \in \mathbb{R}^{n \times (p+1)}$ 是非随机且列满秩的

Problem 1

Given conditions:

- **(A1)** The relationship between response (y) and covariates (X) is linear;
- **(A2)** X is a non-stochastic matrix and $\operatorname{rank}(X) = p$;
- **(A3)** $\mathbb{E}(\varepsilon) = 0$. This implies $\mathbb{E}(y) = X\beta$;
- **(A4)** $\operatorname{Cov}(\varepsilon) = \mathbb{E}(\varepsilon\varepsilon^\top) = \sigma^2 I_n$ (Homoscedasticity);
- **(A5)** ε follows a multivariate normal distribution $N(0, \sigma^2 I_n)$ (Normality).

Part (1)

Prove that the LSE estimator $\hat{\beta}_{\text{LSE}} = (X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}y$ is the same as the maximum likelihood estimator.

Solution:
考虑一般的多元线性回归模型 $y = X\beta + \varepsilon$
(我们这里不对 ε 做任何统计上的假设，但假设 $X \in \mathbb{R}^{n \times (p+1)}$ 是非随机且列满秩的)
则我们有:

$$\hat{\beta}_{\text{LSE}} := \arg\min_{\beta \in \mathbb{R}^{p+1}} \|y - X\beta\|_2^2$$

求解这个无约束凸优化问题得到 $\hat{\beta}_{\text{LSE}} = (X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}y$

现在我们考虑条件正态模型 (即补充假设 $\varepsilon \sim N(0_n, \sigma^2 I_n)$)，则我们有 $y \sim N(X\beta, \sigma^2 I_n)$
于是我们可以逐步推导对数似然函数 $\log L(\beta, \sigma^2 | X, y)$:

$$\begin{aligned} f(y) &= \frac{1}{(\sqrt{2\pi})^n |\sigma^2 I_n|^{\frac{1}{2}}} \exp\{-\frac{1}{2}(y - X\beta)^{\mathrm{T}}(\sigma^2 I_n)^{-1}(y - X\beta)\} \\ &= \frac{1}{(\sqrt{2\pi}\sigma)^n} \exp\{-\frac{1}{2\sigma^2}(y - X\beta)^{\mathrm{T}}(y - X\beta)\} \end{aligned}$$

$$L(\beta, \sigma^2 | X, y) = f(y) = \frac{1}{(\sqrt{2\pi}\sigma)^n} \exp\{-\frac{1}{2\sigma^2}(y - X\beta)^{\mathrm{T}}(y - X\beta)\}$$

$$\log L(\beta, \sigma^2 | X, y) = -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log(\sigma^2) - \frac{1}{2\sigma^2}(y - X\beta)^{\mathrm{T}}(y - X\beta)$$

最大化对数似然函数，我们有:

$$
\begin{aligned}
\hat{\beta}_{\text{MLE}} &:= \arg\max_{\beta\in\mathbb{R}^{p+1}} \log L(\beta,\sigma^2|X,y) \\
&= \arg\max_{\beta\in\mathbb{R}^{p+1}} \left\{-\frac{n}{2}\log{(2\pi)} - \frac{n}{2}\log{(\sigma^2)} - \frac{1}{2\sigma^2}(y-X\beta)^{\mathrm{T}}(y-X\beta)\right\} \\
&= \arg\min_{\beta\in\mathbb{R}^{p+1}} \|y-X\beta\|_2^2 \\
&= \hat{\beta}_{\text{LSE}} \\
&= (X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}y
\end{aligned}
$$

Part (2)

Prove the following results:

- ① $\hat{\beta}\sim N(\beta,\sigma^2(X^{\mathrm{T}}X)^{-1})$
- ② $(n-p-1)s^2\sim\sigma^2\chi^2_{n-p-1}$

Solution:
在条件正态模型 $y\sim N(X\beta,\sigma^2I_n)$ 和 $X\in\mathbb{R}^{n\times(p+1)}$ 非随机且列满秩的假设下，
我们知道 $\hat{\beta}=(X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}y$ 作为 y 的线性组合也是服从多元正态的:

$$
\begin{aligned}
\hat{\beta}=(X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}y &\sim N((X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}X\beta,[(X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}]^{\mathrm{T}}\sigma^2I_n(X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}) \\
&= N(\beta,\sigma^2(X^{\mathrm{T}}X)^{-1})
\end{aligned}
$$

下面我们证明 $\frac{n-p-1}{\sigma^2}s^2=\frac{1}{\sigma^2}\varepsilon^{\mathrm{T}}(I_n-H)\varepsilon$ 服从自由度为 $n-p-1$ 的卡方分布 $\chi^2_{(n-p-1)}$

投影矩阵 $H=X(X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}$ 具有以下几个性质:

- ① 对称: $H^{\mathrm{T}}=H$ (进而有 $(I_n-H)^{\mathrm{T}}=I_n-H$)
  这保证了 H 的 n 个特征值均为实数 (进而 I_n-H 的 n 个特征值也均为实数)
- ② 幂等: $H^2=H$ (进而有 $(I_n-H)^2=I_n-H$)
  这保证了 H 的特征值只能是 0 或 1 (进而 I_n-H 的 n 个特征值也只能是 0 或 1)
- ③ 迹与特征值: $\operatorname{tr}(H)=p+1$ (进而有 $\operatorname{tr}(I_n-H)=n-p-1$)
  这保证了 H 的特征值为 $n-p-1$ 个 0 和 $p+1$ 个 1 (进而 I_n-H 的特征值为 $n-p-1$ 个 1 和 $p+1$ 个 0)

设 I_n-H 的谱分解 (实对称阵一定具有谱分解) 为:

$$
U^H(I_n-H)U=\Lambda=\operatorname{diag}\{\underbrace{1,\dots,1}_{n-p-1},\underbrace{0,\dots,0}_{p+1}\}
$$

记 $\eta:=\frac{1}{\sigma}U^H\varepsilon$，根据 $\varepsilon\sim N(0_n,\sigma^2I_n)$ 可知
$\eta=\frac{1}{\sigma}U^H\varepsilon\sim N(\frac{1}{\sigma}U^H0_n,\frac{1}{\sigma^2}U^H\sigma^2I_nU)=N(0_n,I_n)$
这表明 $\eta=[\eta_1,\dots,\eta_n]^{\mathrm{T}}$ 的分量是独立同分布的标准正态随机变量.
于是我们有:

$$
\begin{aligned}
\frac{n-p-1}{\sigma^2}s^2 &= \frac{1}{\sigma^2}\varepsilon^{\mathrm{T}}(I_n-H)\varepsilon \\
&= \frac{1}{\sigma^2}\varepsilon^{\mathrm{T}}U\Lambda U^H\varepsilon \\
&= \eta^{\mathrm{T}}\Lambda\eta \\
&= \sum_{i=1}^{n-p-1}\eta_i^2\sim\chi^2_{(n-p-1)} \quad (\text{note that } \Lambda=\operatorname{diag}\{\underbrace{1,\dots,1}_{n-p-1},\underbrace{0,\dots,0}_{p+1}\})
\end{aligned}
$$

这样我们就证明了 $\frac{n-p-1}{\sigma^2}s^2 = \frac{1}{\sigma^2}\varepsilon^{\mathrm{T}}(I_n - H)\varepsilon$ 服从自由度为 $n-p-1$ 的卡方分布 $\chi^2_{(n-p-1)}$
即有 $(n-p-1)s^2 \sim \sigma^2\chi^2_{n-p-1}$

Problem 2

Suppose y follows the log-linear regression relationship with $x \in \{1\} \times \mathbb{R}^p$, i.e.,
$\log(y) = x^{\mathrm{T}}\beta + \varepsilon$
where ε follows normal distribution $N(0, \sigma^2)$.
Please calculate $\mathbb{E}(y)$.

> **Lemma: 正态随机变量的矩母函数**
> 正态随机变量 $X \sim \mathrm{N}(\mu, \sigma^2)$ 有 $f(x) = \frac{1}{\sqrt{2\pi}\sigma}\exp\{-\frac{(x-\mu)^2}{2\sigma^2}\}$ $(\forall\, x \in \mathbb{R})$ 且
> $\begin{cases} \mathrm{E}(X) = \mu \\ \mathrm{E}(X^2) = \sigma^2 + \mu^2 \\ \mathrm{Var}(X) = \sigma^2 \end{cases}$
> 考虑标准正态随机向量 $Z \sim \mathrm{N}(0, 1)$，其矩母函数为:
> $$\begin{aligned} M_X(t) &= \mathrm{E}[e^{tX}] \\ &= \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} e^{tx} \cdot e^{-x^2/2}\mathrm{d}x \\ &= \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} e^{-(x^2-2tx)/2}\mathrm{d}x \\ &= e^{t^2/2}\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} e^{-(x-t)^2/2}\mathrm{d}(x-t) \\ &= e^{t^2/2} \cdot 1 \\ &= e^{t^2/2} \end{aligned}$$
> 则 $X = \sigma Z + \mu \sim \mathrm{N}(\mu, \sigma^2)$ 的矩母函数为:
> $$\begin{aligned} M_X(t) &= \mathrm{E}[e^{tX}] \\ &= \mathrm{E}[e^{t(\sigma Z + \mu)}] \\ &= e^{t\mu}\mathrm{E}[e^{t\sigma Z}] \\ &= \exp\{\frac{\sigma^2 t^2}{2} + \mu t\} \end{aligned}$$

Solution:

$$\begin{aligned} \mathrm{E}[y] &= \mathrm{E}[\exp\left(x^{\mathrm{T}}\beta + \varepsilon\right)] \\ &= \exp(x^{\mathrm{T}}\beta) \cdot \mathrm{E}[\exp(\varepsilon)] \quad (\text{note that } \varepsilon \sim N(0, \sigma^2) \text{ so that } M_\varepsilon(t) = \mathrm{E}[e^{t\varepsilon}] = \exp\{\frac{\sigma^2 t^2}{2}\}) \\ &= \exp(x^{\mathrm{T}}\beta) \cdot \exp\{\frac{\sigma^2 \cdot 1^2}{2}\} \\ &= \exp\{x^{\mathrm{T}}\beta + \frac{\sigma^2}{2}\} \end{aligned}$$

Problem 3

Define $\hat{y} = X\beta$.
Let the intercept be included in the regression model.
Define the total sum of squares (TSS), explained sum of squares (ESS) and residual sum of squares (RSS) as follows:

$$\begin{aligned}\mathrm{TSS} &= \|y - \bar{y}1_n\|_2^2 \\ \mathrm{ESS} &= \|\hat{y} - \bar{y}1_n\|_2^2 \\ \mathrm{RSS} &= \|y - \bar{y}\|_2^2\end{aligned}$$

Please prove: $\mathrm{TSS} = \mathrm{ESS} + \mathrm{RSS}$

Solution:

$$\begin{aligned}\mathrm{TSS} &= \|y - \bar{y}1_n\|_2^2 \\ &= (y - \bar{y}1_n)^{\mathrm{T}}(y - \bar{y}1_n) \\ &= y^{\mathrm{T}}y - n\bar{y}^2 \\ &= y^{\mathrm{T}}y - n(\frac{1}{n}1_n^{\mathrm{T}}y)^2 \\ &= y^{\mathrm{T}}(I_n - \frac{1}{n}1_n1_n^{\mathrm{T}})y \\ \hline \mathrm{ESS} &= \|\hat{y} - \bar{y}1_n\|_2^2 \\ &= \|Hy - \frac{1}{n}1_n^{\mathrm{T}}y1_n\|_2^2 \\ &= \|(H - \frac{1}{n}1_n1_n^{\mathrm{T}})y\|_2^2 \\ &= y^{\mathrm{T}}(H - \frac{1}{n}1_n1_n^{\mathrm{T}})^{\mathrm{T}}(H - \frac{1}{n}1_n1_n^{\mathrm{T}})y \quad (\text{note that } H \text{ satisfies} \begin{cases} H^{\mathrm{T}} = H \\ H^2 = H \\ H1_n = 1_n \end{cases}) \\ &= y^{\mathrm{T}}(H - \frac{1}{n}1_n1_n^{\mathrm{T}})y \\ \hline \mathrm{RSS} &= \|y - \hat{y}\|_2^2 \\ &= \|y - Hy\|_2^2 \\ &= y^{\mathrm{T}}(I_n - H)y \\ &= y^{\mathrm{T}}[(I_n - \frac{1}{n}1_n1_n^{\mathrm{T}}) - (H - \frac{1}{n}1_n1_n^{\mathrm{T}})]y \\ &= y^{\mathrm{T}}(I_n - \frac{1}{n}1_n1_n^{\mathrm{T}})y - y^{\mathrm{T}}(H - \frac{1}{n}1_n1_n^{\mathrm{T}})y \\ &= \mathrm{TSS} - \mathrm{ESS}\end{aligned}$$

因此我们有 $\mathrm{TSS} = \mathrm{ESS} + \mathrm{RSS}$

Problem 4

(岭回归分析)
在实际问题中，我们常常会遇到样本容量相对较小，而特征很多的场景，此时会出现多重共线性和过拟合的问题.
为缓解这些问题，常在线性回归的损失函数中引入正则化项 $\mathrm{punish}(\beta)$:

$$\hat{\beta}_{\text{punished}} := \arg\min_{\beta \in \mathbb{R}^{p+1}} \left\{\|y - X\beta\|_2^2 + \lambda \cdot \mathrm{punish}(\beta)\right\}$$

其中 p 为解释变量的个数，$X \in \mathbb{R}^{n\times(p+1)}$ 为设计矩阵，$\beta \in \mathbb{R}^{p+1}$ 为回归参数，$\lambda > 0$ 为正则化系数.
正则化表示了对模型的一种偏好，希望在保持良好预测性能的同时，选择较为简单的模型，从而提高模型的泛化能力.
当 $\mathrm{punish}(\beta) = \|\beta\|_2^2$ 时，即为岭回归问题:

$$\hat{\beta}_{\mathrm{Ridge}} := \arg\min_{\beta\in\mathbb{R}^{p+1}} \left\{\|y - X\beta\|_2^2 + \lambda\|\beta\|_2^2\right\}$$

Part (1)

试证明 $\hat{\beta}_{\mathrm{Ridge}}$ 的显式解具有以下等价形式:

$$\hat{\beta}_{\mathrm{Ridge}} = (X^{\mathrm{T}}X + \lambda I_{p+1})^{-1}X^{\mathrm{T}}y = X^{\mathrm{T}}(XX^{\mathrm{T}} + \lambda I_n)^{-1}y$$

并分析上述形式分别在什么情况下计算速度更快?

Solution:
记目标函数 $f(\beta) := \|y - X\beta\|_2^2 + \lambda\|\beta\|_2^2$，则我们有:

$$\begin{aligned}\nabla f(\beta) &= -2X^{\mathrm{T}}(y - X\beta) + 2\lambda\beta \\ \nabla^2 f(\beta) &= 2X^{\mathrm{T}}X + \lambda I_{p+1} \succ 0\end{aligned}$$

因此这是一个无约束凸优化问题，全局最小点即为驻点.
令 $\nabla f(\beta) = 0_{p+1}$ 即得 $\hat{\beta}_{\mathrm{Ridge}} = (X^{\mathrm{T}}X + \lambda I_{p+1})^{-1}X^{\mathrm{T}}y$

$$\begin{aligned}\hat{\beta}_{\mathrm{Ridge}} &= (X^{\mathrm{T}}X + \lambda I_{p+1})^{-1}X^{\mathrm{T}}y \\ &= (X^{\mathrm{T}}X + \lambda I_{p+1})^{-1}X^{\mathrm{T}}(XX^{\mathrm{T}} + \lambda I_n)(XX^{\mathrm{T}} + \lambda I_n)^{-1}y \\ &= (X^{\mathrm{T}}X + \lambda I_{p+1})^{-1}(X^{\mathrm{T}}X + \lambda I_{p+1})X^{\mathrm{T}}(XX^{\mathrm{T}} + \lambda I_n)^{-1}y \\ &= X^{\mathrm{T}}(XX^{\mathrm{T}} + \lambda I_n)^{-1}y\end{aligned}$$

这样我们就得到了 $\hat{\beta}_{\mathrm{Ridge}}$ 的两种等价的计算公式:

$$\hat{\beta}_{\mathrm{Ridge}} = (X^{\mathrm{T}}X + \lambda I_{p+1})^{-1}X^{\mathrm{T}}y = X^{\mathrm{T}}(XX^{\mathrm{T}} + \lambda I_n)^{-1}y$$

前者的计算复杂度主要来源于矩阵乘积 $X^{\mathrm{T}}X$ 和逆矩阵 $(X^{\mathrm{T}}X + \lambda I_{p+1})^{-1}$ 的计算，为 $O((p+1)^2n) + O((p+1)^3)$
后者的计算复杂度主要来源于矩阵乘积 XX^{T} 和逆矩阵 $(XX^{\mathrm{T}} + \lambda I_n)^{-1}$ 的计算，为 $O(n^2(p+1)) + O(n^3)$
因此当 $p+1 < n$ 时，前者的计算更高效;
而当 $p+1 > n$ 时，后者的计算更高效;
当 $p+1 = n$ 时，二者的计算速度相近.

Part (2)

分析岭回归估计量 $\hat{\beta}_{\mathrm{Ridge}}$ 与最小二乘估计量 $\hat{\beta}_{\mathrm{LSE}}$ 的区别.

Solution:
岭回归估计量 $\hat{\beta}_{\mathrm{Ridge}}$ 的适用范围比最小二乘估计量 $\hat{\beta}_{\mathrm{LSE}}$ 更广:
最小二乘估计量 $\hat{\beta}_{\mathrm{LSE}} = (X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}y$ 对设计矩阵 $X \in \mathbb{R}^{n\times(p+1)}$ 的要求是 $\mathrm{rank}(X) = p+1 \leq n$
这保证了 $X^{\mathrm{T}}X$ 是正定矩阵，因而可逆.
而岭回归估计量 $\hat{\beta}_{\mathrm{Ridge}}$ 对 X 的形状和秩没有要求，
任何情况下都能通过以下两种方式计算 (尽管可能存在计算速度的差别):

$$\hat{\beta}_{\text{Ridge}} = (X^{\mathrm{T}}X + \lambda I_{p+1})^{-1}X^{\mathrm{T}}y = X^{\mathrm{T}}(XX^{\mathrm{T}} + \lambda I_n)^{-1}y$$

从某种角度来说，岭回归估计量 $\hat{\beta}_{\text{Ridge}}$ 是最小二乘估计量 $\hat{\beta}_{\text{LSE}}$ 的推广.
当 $\text{rank}(X) = p+1 \leq n$ 时，取 $\lambda = 0$ 的岭回归估计量 $\hat{\beta}_{\text{Ridge}}$ 即为最小二乘估计量 $\hat{\beta}_{\text{LSE}}$

$\hat{\beta}_{\text{Ridge}}$ 具有收缩性，当 $\lambda \to \infty$ 时 $\hat{\beta}_{\text{Ridge}}$ 单调递减趋于 0

Problem 5

北京租房数据集介绍:
本案例的数据来源于某租房平台，数据已被划分为训练集和测试集，分别对应文件 `train_data.csv` 和 `test_data.csv`
数据集共采集了北京市某年某月 5149 条合租房源的信息.
本案例针对合租房间进行分析，若同一套房中有多个待租的房间， 这些房间在本案例的数据中会对应多条数据，
每一条数据对应其中一个合租房间，并且这些房间的数据中房源整体的信息相同 (如房屋结构、地理位置等)，
但租赁面积和月租金不同.
具体数据说明表如下:

变量类型		变量名		详细说明	取值范围
因变量		rent	季均销量	定量变量，单位：元	1150~6460
自变量	内部结构	area	租赁房间面积	定量变量，单位：平方米	5~30
		room	租赁房间类型	定性变量，2 个水平	主卧、次卧
		bedroom	卧室数	定量变量，单位：个	2~5
		livingroom	厅数	定量变量，单位：个	1~2
		bathroom	卫生间数	定量变量，单位：个	1~2
		heating	供暖方式	定性变量，2 个水平	集中供暖、自采暖
	外部条件	floor_grp	所在楼层	定性变量，3 个水平	高楼层、中楼层、低楼层
		subway	邻近地铁	定性变量，2 个水平	是、否
		region	所在城区	定性变量，11 水平	朝阳、海淀、东城、西城、昌平、大兴、通州、石景山、丰台、顺义、房山

- **因变量**:
  - **rent**: 季均销量，定量变量，单位: 元，取值范围: 1150~6460.
- **自变量 - 内部结构**:
  - **area**: 租赁房间面积，定量变量，单位: 平方米，取值范围: 5~30.
  - **room**: 租赁房间类型，定性变量，2个水平: 主卧, 次卧.
  - **bedroom**: 卧室数，定性变量，单位: 个，取值范围: 2~5.
  - **livingroom**: 厅数，定性变量，单位: 个，取值范围: 1~2.
  - **bathroom**: 卫生间数，定性变量，单位: 个，取值范围: 1~2.
  - **heating**: 供暖方式，定性变量，2个水平: 集中供暖, 自采暖.
- **自变量 - 外部条件**:
  - **floor_grp**: 所在楼层，定性变量，3个水平: 高楼层, 中楼层, 低楼层.
  - **subway**: 邻近地铁，定性变量，2个水平: 是, 否.

- **region**: 所在城区，定性变量，11个水平: 朝阳, 海淀, 东城, 西城, 昌平, 大兴, 通州, 石景山, 丰台, 顺义, 房山.

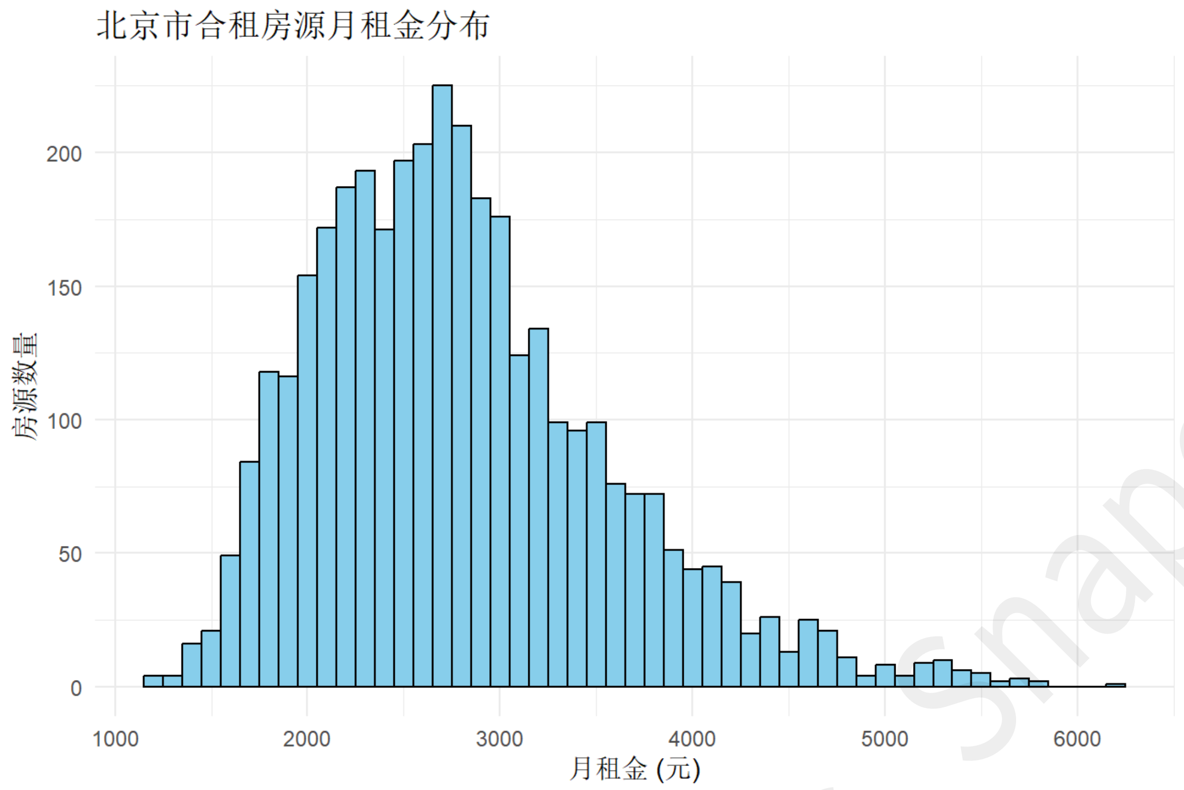
针对北京租房数据，完成以下任务:
提示: 将属性变量转化为 onehot 编码后再利用显式解得到参数估计.

Part (1)

完成数据读入与汇总统计，绘制训练集数据中月租金 (rent) 的直方图，观察月租金的大致分布，并进行简要解读.
绘制训练集数据中月租金 (rent)-城区 (region) 分组箱线图，分析不同城区的房价差异，并给出简要解读.

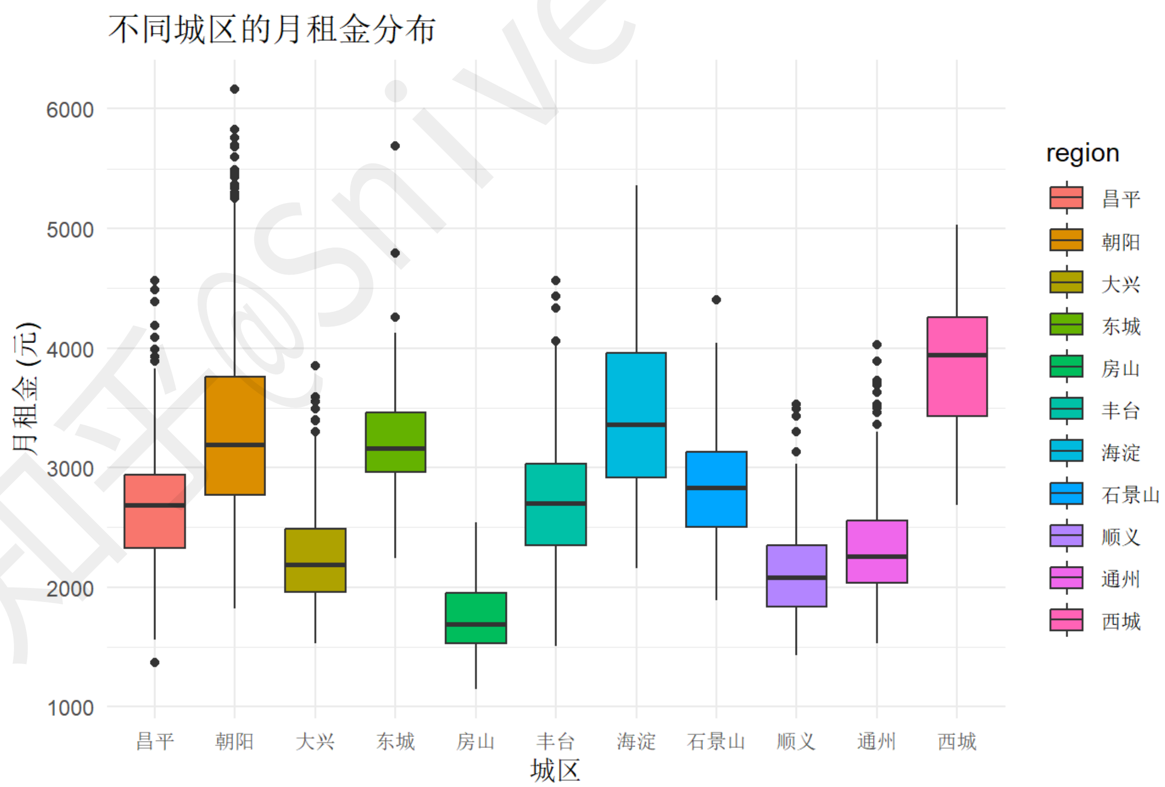
Solution:

```r
# 加载所需的库
library(ggplot2)
library(dplyr)

# 读取训练集数据
train_data <- read.csv("train_data.csv")

# 查看数据的基本结构
str(train_data)

# 数据汇总统计
summary(train_data)

# 绘制月租金(rent)的直方图
ggplot(train_data, aes(x = rent)) +
  geom_histogram(binwidth = 100, fill = "skyblue", color = "black") +
  labs(title = "北京市合租房源月租金分布",
       x = "月租金 (元)",
       y = "房源数量") +
  theme_minimal() # 美化图表

# 绘制月租金(rent) - 城区(region)的分组箱线图
ggplot(train_data, aes(x = region, y = rent)) +
  geom_boxplot(aes(fill = region)) +
  labs(title = "不同城区的月租金分布",
       x = "城区",
       y = "月租金 (元)") +
  # 将 x 轴的文本旋转 45 度，避免文字重叠
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  theme_minimal() # 美化图表
```



从图中可以观察到房源数量关于月租金 `rent` 的**右偏分布**

房源大部分集中在较低的月租金范围，随着月租金的增加，房源数量逐渐减少，呈现出长尾的右偏趋势.
这表明大部分租客的预算相对有限，因而市场上经济适用的房源较多;
高端合租房源在市场中较为稀缺，且高租金房源的需求相对较少.
合租房源的租户群体，可能为刚毕业的年轻人或收入较低的人群，因此价格成为主要考虑因素，导致低租金房源需求旺盛.



从图中可以看出月租金价格从低到高的顺序约为:

- ① 房山、顺义、大兴、通州 $(1700 \sim 2500)$
- ② 昌平、丰台、石景山 $(2500 \sim 3000)$
- ③ 东城、朝阳、海淀、西城 $(3000 \sim 4000)$

同时我们发现各地区月租金价格的中位数普遍靠近 25% 分位数，这表明市场以经济适用房源为主.
离群值普遍位于高于 75% 分位数的范围.
离群值的普遍存在表明高租金房源虽然数量较少，但在市场中仍然存在一定的需求，且其价格明显高于大多数房源.

Part (2)

利用训练集建立以月租金 (rent) 为因变量，其余为自变量的线性回归模型，编程实现最小二乘估计 (不调用回归分析的包)，
写出拟合得到的模型并计算测试集上的均方误差 (Mean Square Error, MSE)

Solution:
数据预处理:

```
# 读取训练集和测试集数据
train_data <- read.csv("train_data.csv")
test_data <- read.csv("test_data.csv")

# 数据预处理
# 将分类变量转换为因子
# 在 R 的因子变量中，会选择一个水平作为参考水平 (通常是第一个水平)
# 并不为该水平创建虚拟变量. 这可以避免冗余 (即多重共线性) 问题.
# 为保证参考水平统一，故先将训练集和测试集合并
combined_data <- rbind(train_data, test_data)
combined_data$room <- factor(combined_data$room,
                             levels = c("次卧", "主卧"))
combined_data$heating <- factor(combined_data$heating,
                                levels = c("自采暖", "集中供暖"))
combined_data$floor_grp <- factor(combined_data$floor_grp,
                                  levels = c("低楼层", "中楼层", "高楼层"))
combined_data$subway <- factor(combined_data$subway,
                               levels = c("否", "是"))
combined_data$region <- factor(combined_data$region,
                               levels = c("房山", "顺义", "大兴", "通州", "昌平",
"丰台", "石景山", "东城", "朝阳", "海淀", "西城"))

# 重新分割数据
train_data <- combined_data[1:nrow(train_data), ]
test_data <- combined_data[(nrow(train_data) + 1):nrow(combined_data), ]

# 将因变量与自变量分开
y_train <- train_data$rent
X_train <- model.matrix(~ . - rent, data = train_data)
```

最小二乘估计:

```
# 最小二乘法估计
# 计算回归系数: beta = (X'X)^(-1)X'y
X_transpose <- t(X_train)
beta <- solve(X_transpose %*% X_train) %*% (X_transpose %*% y_train)

# 拟合模型的公式
model_formula <- paste("rent = ", paste(beta[-1], " * ", colnames(X_train)[-1],
collapse = " + "), sep = "")
```

```
cat("拟合得到的模型:\n", model_formula, "\n")

# 在测试集上进行预测
y_test <- test_data$rent
X_test <- model.matrix(~ . - rent, data = test_data)
y_pred <- X_test %*% beta

# 计算均方误差 (MSE)
mse <- mean((y_test - y_pred)^2)
cat("测试集上的均方误差 (MSE):", mse, "\n")
```

拟合得到的模型:

```
rent =
-101.752247288543  *  bedroom
+ -212.558098885062  *  livingroom
+ 207.196383568542  *  bathroom
+ 76.2925453280663  *  area
+ 17.5391176899606  *  room主卧
+ -47.7109896036935  *  floor_grp中楼层 + -3.63764513735526  *  floor_grp高楼层
+ 279.644619386072  *  subway是
+ 377.343954815613  *  region顺义 + 391.166262161219  *  region大兴 +
424.546005319518  *  region通州
+ 863.57243150158  *  region昌平 + 914.938194276876  *  region丰台 +
806.077782059639  *  region石景山
+ 1404.30236275302  *  region东城 + 1444.8445786761  *  region朝阳 +
1705.69610344992  *  region海淀
+ 1809.1900892034  *  region西城
+ 161.763807705411  *  heating集中供暖
```

测试集上的**均方误差** (MSE) 为 218413.5

Part (3)

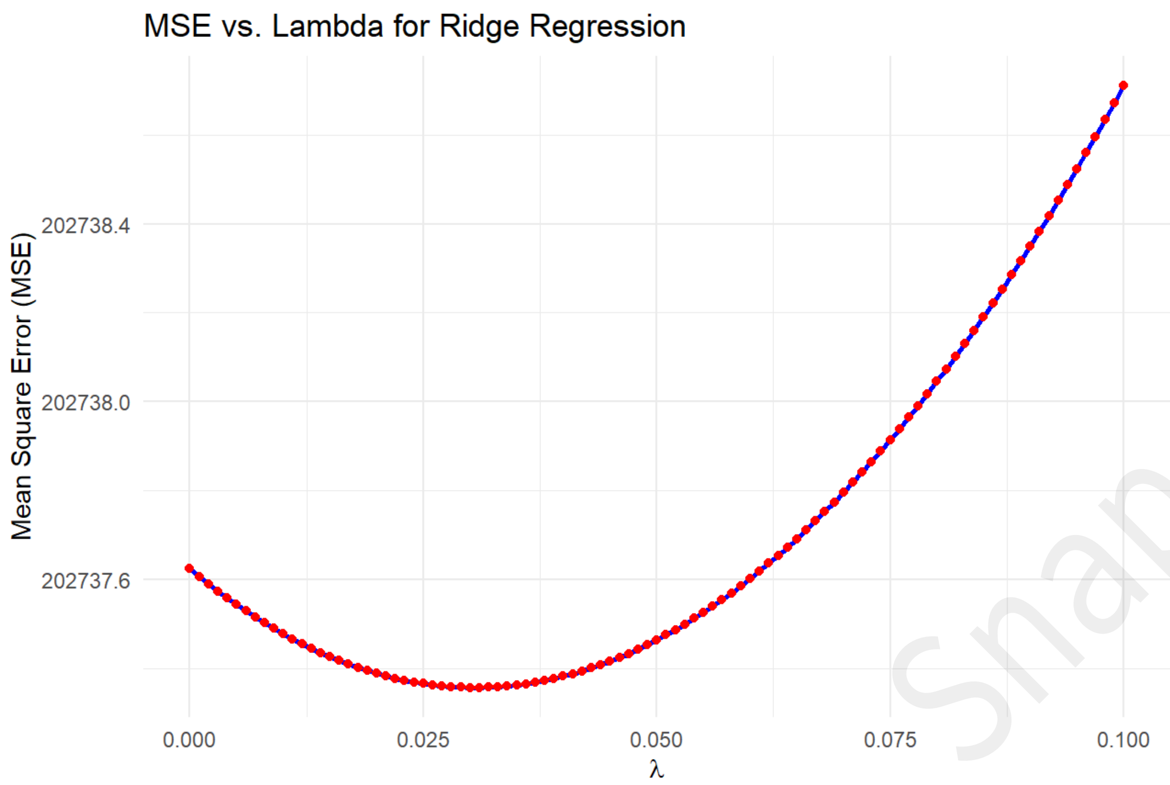
编程实现岭回归估计 (不调用回归分析的包)，在训练集上使用十折交叉验证，
画出验证集上平均均方误差 (Mean Square Error, MSE) 与 λ 的折线图，选出合适的 λ

Solution:
十折交叉验证的岭回归估计:

```
# 定义岭回归函数
ridge_regression <- function(X, y, lambda) {
  p <- ncol(X)
  identity_matrix <- diag(p)
  beta_ridge <- solve(t(X) %*% X + lambda * identity_matrix) %*% (t(X) %*% y)
  return(beta_ridge)
}

# 十折交叉验证
set.seed(51)
k <- 10
folds <- cut(seq(1, nrow(X_train)), breaks = k, labels = FALSE)

# 定义待测试的 lambda 值
lambda_values <- seq(0, 0.1, by = 0.001)
```

```r
mse_values <- numeric(length(lambda_values))

for (i in seq_along(lambda_values)) {
  lambda <- lambda_values[i]
  mse_fold <- numeric(k)

  for (j in 1:k) {
    # 生成训练集和验证集
    test_indices <- which(folds == j, arr.ind = TRUE)
    X_train_fold <- X_train[-test_indices, ]
    y_train_fold <- y_train[-test_indices]
    X_test_fold <- X_train[test_indices, ]
    y_test_fold <- y_train[test_indices]

    # 进行岭回归估计
    beta_ridge <- ridge_regression(X_train_fold, y_train_fold, lambda)

    # 进行预测
    y_pred_fold <- X_test_fold %*% beta_ridge
    mse_fold[j] <- mean((y_test_fold - y_pred_fold)^2)
  }

  mse_values[i] <- mean(mse_fold)  # 计算每个 lambda 的平均 MSE
}

# 绘制 MSE 与 lambda 的折线图
mse_df <- data.frame(lambda = lambda_values, mse = mse_values)

ggplot(mse_df, aes(x = lambda, y = mse)) +
  geom_line(color = "blue", linewidth = 1) +
  geom_point(color = "red") +
  labs(title = "MSE vs. Lambda for Ridge Regression",
       x = expression(lambda),
       y = "Mean Square Error (MSE)") +
  theme_minimal()

# 找到最小的 MSE 及其对应的 lambda
best_lambda <- lambda_values[which.min(mse_values)]
cat("最佳的 lambda 值为:", best_lambda, "\n")
cat("最佳的均方误差为:", min(mse_values), "\n")
```

最佳的 λ 值为: 0.031
最佳的均方误差为: 202737.4



Part (4)

用选出的 λ 在训练集拟合最终模型，写出拟合得到的模型并计算测试集上的均方误差.

Solution:
使用选取的 $\lambda = 0.031$ 进行岭回归估计:

```
# 使用选取的 λ = 0.031 进行岭回归估计
lambda <- 0.031
beta_ridge <- ridge_regression(X_train, y_train, lambda)

# 拟合模型的公式
model_formula <- paste("rent = ", paste(beta_ridge[-1], " * ", colnames(X_train)
[-1], collapse = " + "), sep = "")
cat("拟合得到的模型:\n", model_formula, "\n")

# 在测试集上进行预测
y_test <- test_data$rent
X_test <- model.matrix(~ . - rent, data = test_data)
y_pred <- X_test %*% beta_ridge

# 计算均方误差 (MSE)
mse <- mean((y_test - y_pred)^2)
cat("测试集上的均方误差 (MSE):", mse, "\n")
```

拟合得到的模型:

```
rent =
-101.756828165574  *  bedroom
+ -211.553477876667  *  livingroom
+ 207.195877013351  *  bathroom
+ 76.3094054950661  *  area
+ 17.4536133911951  *  room主卧
+ -47.658765537249  *  floor_grp中楼层 + -3.61497115452346  *  floor_grp高楼层
+ 279.613077658313  *  subway是
+ 374.996276966323  *  region顺义 + 388.87486215869  *  region大兴 +
422.264934231048  *  region通州
+ 861.273830053909  *  region昌平 + 912.640392332159  *  region丰台 +
803.663216079399  *  region石景山
+ 1401.41105009289  *  region东城 + 1442.56564623108  *  region朝阳 +
1703.30741505259  *  region海淀
+ 1806.05333990126  *  region西城
+ 161.979078159631  *  heating集中供暖
```

测试集上的均方误差 (MSE): 218402.3

The End