6.1 Problem 1

(a) Define \( g(y) = T(y) - Ly \). Then, we have that
\[
\mathbb{E} \left[ \| T(y) - L\mu \|_2^2 \right] = \mathbb{E} \left[ \| g(y) + Ly - L\mu \|_2^2 \right]
\]
\[
= \mathbb{E} \left[ \| Ly - L\mu \|_2^2 \right] + \mathbb{E} \left[ \| g(y) \|_2^2 \right] + 2\mathbb{E} \left[ (Ly - L\mu)^\top g(y) \right]
\]
\[
= \mathbb{E} \left[ (y - \mu)^\top L^\top L(y - \mu) \right] + \mathbb{E} \left[ \| g(y) \|_2^2 \right] + 2\mathbb{E} \left[ (y - \mu)^\top L^\top g(y) \right].
\]

Now note that
\[
\mathbb{E} \left[ (y - \mu)^\top L^\top L(y - \mu) \right] = \mathbb{E} \left[ \text{Tr}(L^\top L(y - \mu)(y - \mu)^\top) \right]
\]
\[
= \text{Tr} \left( L^\top \mathbb{E}[(y - \mu)(y - \mu)^\top) \right)
\]
\[
= \text{Tr} \left( L^\top L \right).
\]

Also, define \( h(y) = L^\top g(y) \). Then, for each \( i \), we have that (where \( \varphi \) is the density of a standard normal distribution)
\[
\mathbb{E}[(y - \mu)^\top h_i(y) | y_1, \ldots, y_{i-1}, y_{i+1}, \ldots, y_n]
\]
\[
= \int (y - \mu)^\top h_i(y) \varphi(y_i - \mu_i) dy_i
\]
\[
= - \int h_i(y) \frac{\partial \varphi}{\partial y_i}(y_i - \mu_i) dy_i
\]
\[
= - [h_i(y) \varphi(y_i - \mu_i)]_{-\infty}^{+\infty} + \int \frac{\partial h_i(y)}{\partial y_i} \varphi(y_i - \mu_i) dy_i
\]
\[
= \mathbb{E} \left[ \frac{\partial h_i(y)}{\partial y_i} | y_1, \ldots, y_{i-1}, y_{i+1}, \ldots, y_n \right].
\]

(i.e., the useful result \( E[Gf(G)] = E[f'(G)] \) for \( G \sim N(0, 1) \)). Note that we assumed in the second to last line that \( h_i(y) \varphi(y_i - \mu_i) \) converges to 0 as \( y_i \) goes to \( \pm \infty \) for all fixed \( y_1, \ldots, y_{i-1}, y_{i+1}, \ldots, y_n \). We also assumed that \( g \) and hence \( h \) is almost differentiable. Therefore,
\[
\mathbb{E}[(y - \mu)^\top h(y)] = \mathbb{E} \left[ \sum_{i=1}^n \frac{\partial h_i(y)}{\partial y_i} \right].
\]
We conclude that
\[ \mathbb{E} \left[ \|T(y) - L\mu\|^2 \right] = \text{Tr} (L^\top L) + \mathbb{E} \left[ \|g(y)\|^2_2 + 2 \sum_{i=1}^{n} \frac{\partial h_i(y)}{\partial y_i} \right], \]
and an unbiased estimate for the MSE is
\[
\hat{E} = \text{Tr} (L^\top L) + \|g(y)\|^2_2 + 2 \sum_{i=1}^{n} \frac{\partial h_i(y)}{\partial y_i}.
\]

(b) We know that for any positive definite matrix \( \Sigma \) we can decompose it as \( \Sigma = U^\top D^2 U \), where \( U \) is an orthogonal matrix and \( D \) is a diagonal matrix. Then define \( L = U^\top DU \) and \( x = L^{-1}y \). Hence, \( x \sim N(L^{-1}\mu, I) \). Denote \( \mu' = L^{-1}\mu \).

Let \( T'(x) = T(Lx) \). Then we want to find an unbiased estimator of
\[ \mathbb{E} \left[ \|T'(x) - L\mu'\|^2 \right] = \mathbb{E} \left[ \|T(y) - \mu\|^2 \right]. \]
Applying the result of part (a), we get the following estimator
\[
\hat{E} = \text{Tr} (L^\top L) + \|g(x)\|^2_2 + 2 \sum_{i=1}^{n} \frac{\partial h_i(x)}{\partial x_i},
\]
where we recall that \( x = L^{-1}y \). We can further simplify
\[
\sum_{i,k=1}^{n} L_{ki} \frac{\partial T_k(y)}{\partial x_i} = \sum_{i,k,l=1}^{n} L_{ki} L_{li} \frac{\partial T_k(y)}{\partial y_l} = \sum_{i,k=1}^{n} \Sigma_{lk} \frac{\partial T_k(y)}{\partial y_l}.
\]
Hence
\[
\hat{E} = -\text{Tr}(\Sigma) + \|T(y) - y\|^2_2 + 2 \sum_{l,k=1}^{n} \Sigma_{lk} \frac{\partial T_k(y)}{\partial y_l}.
\]

### 6.2 Problem 2

(a) Notice that, on the event \( \{\|X\|^2 > n - 2\}\), we have that \( 1 - (n - 2)/\|X\|^2 \geq 0 \), and so \( \hat{\mu}_{JS} = \hat{\mu}_{JS+} \). Hence, we have
\[
R(\hat{\mu}_{JS}, \mu) - R(\hat{\mu}_{JS+}, \mu) = \mathbb{E}[\|\hat{\mu}_{JS} - \mu\|^2_2 - \|\hat{\mu}_{JS+} - \mu\|^2_2] \mathbf{1}\{\|X\|^2 \leq n - 2\}. \]
(b) Let $X \sim N(\mu, 1)$, with density $f$ for $\mu \neq 0$. Notice that $f$ is symmetric and unimodal around $\mu$ with strictly decreasing tails. Suppose that $\mu_i > 0$. Then, let $x > 0$. If $x \leq \mu$, then we immediately have that $f(-x) < f(x)$. Otherwise, $\mu < x$ in which case we have that

$$f(-x) = f(\mu - (x + \mu)) = f(\mu + (x + \mu)) < f(\mu + (x - \mu)) = f(x).$$

Similarly, if $\mu < 0$, then $f(-x) > f(x)$. In an case, we conclude that

$$\mu(f(x) - f(-x)) > 0.$$ 

Now let $G : \mathbb{R} \to \mathbb{R}_{\geq 0}$ be a function that is symmetric around 0 and strictly positive on some positive-measure set. We then have that

$$\mu \mathbb{E}[G(X)X] = \mu \int_{-\infty}^{\infty} G(x)xf(x)dx$$

$$= \mu \int_{0}^{\infty} G(x)xf(x)dx + \mu \int_{-\infty}^{0} G(x)xf(x)dx$$

$$= \mu \int_{0}^{\infty} G(x)xf(x)dx + \mu \int_{0}^{\infty} G(x)(-xf(-x))dx$$

$$= \int_{0}^{\infty} G(x)x\mu(f(x) - f(-x))dx.$$ 

The integrand is non-negative and strictly positive on a positive-measure set. Hence, $\mu \mathbb{E}[G(X)X] > 0$. We moreover have that, if $G_y$ is a family of such functions indexed by $y \in \mathcal{Y}$ and $Y$ is independent of $X$ with measure $\nu$, then

$$\mu \mathbb{E}[G_y(X)X] = \int_{\mathcal{Y}} \mu \mathbb{E}[G_y(X)X] \nu(dy).$$

Returning to the context of our question, for any $i$, let $X_{-i}$ denote $X$ with the $i$-th entry removed. Since $X$ is a multivariate normal with uncorrelated components, this means that $X_i$ is independent of $X_{-i}$. We have that, taking $G_y(x) = 1\{x^2 + \|y\|_2^2 \geq n - 2\}$, which satisfies our conditions,

$$\mathbb{E}[\mu_iX_i\|X\|_2^2 \leq n - 2] = \mathbb{E}[\mu_iX_i\|X_{-i}\|^2 + X_i^2 \leq n - 2]$$

$$= \mu_i \mathbb{E}[G_{X_{-i}}(X_i)X_i] / \mathbb{P}(\|X\|_2^2 \leq n - 2) > 0.$$
(c) From part (a), we have that
\[
R(\hat{\mu}_{JS}, \mu) - R(\hat{\mu}_{JS+}, \mu) \\
= E[(||\hat{\mu}_{JS} - \mu||^2 - ||\hat{\mu}_{JS+} - \mu||^2)1\{|X||^2 \leq n - 2\}] \\
= E \left[ \left\{ \left( 1 - \frac{n-2}{\|X\|^2} \right) X - \mu \right\}_2^2 - \|\mu\|^2 \right] 1\{|X||^2 \leq n - 2\} \\
= E \left[ \left\{ \left( 1 - \frac{n-2}{\|X\|^2} \right) X \right\}_2^2 - 2 \left( 1 - \frac{n-2}{\|X\|^2} \right) \mu^T X \right] 1\{|X||^2 \leq n - 2\} \\
= E \left[ \left\{ \left( 1 - \frac{n-2}{\|X\|^2} \right) X \right\}_2^2 1\{|X||^2 \leq n - 2\} \right] \\
- 2 \sum_{i=1}^n E \left[ \left( 1 - \frac{n-2}{\|X\|^2} \right) \mu_i X_i 1\{|X||^2 \leq n - 2\} \right]
\]

Note that the random variable in the first expectation is non-negative. As for the second term, we use our result from part (b) with \(G_y(x) = [(n-1)/(x^2 + \|y\|^2) - 1]1\{x^2 + \|y\|^2 \leq n - 2}\) to conclude that
\[
E \left[ \left( \frac{n-2}{\|X\|^2} - 1 \right) \mu_i X_i 1\{|X||^2 \leq n - 2\} \right] = \mu_i E[G_{X_i}(X_i)X_i] > 0.
\]

Thus we have that
\[
R(\hat{\mu}_{JS}, \mu) - R(\hat{\mu}_{JS+}, \mu) > 0,
\]
from which we conclude that the James-Stein estimator is inadmissible.

### 6.3 Problem 3

For this problem, we consider predicting the season average 3 point percentage of NBA players considering their average on the first two months of the season. Realistically, for a player \(i\), if their true season average is \(\mu_i\), we can expect the number \(X_i\) of made shots on the first two months to be distributed according to Bernoulli\((N_i, \mu_i)\), where \(N_i\) is their total number of attempts on these two months. We restrict ourselves to players that have at least 50 3 points attempts on the first two months (and 200 on the whole season), so that realistically we have
\[
X_i/N_i \sim N \left( \mu_i, \frac{\mu_i(1 - \mu_i)}{N_i} \right).
\]

For simplicity, we additionally assume that each \(N_i\) is approximately the same equal to \(\bar{N} \approx 95\). Note that this is a strong assumption, that does not quite hold true in practice.

We then compare two James Stein estimators: one where we assume that each \(\mu_i = \bar{\mu}\), so that each \(\sigma^2 = \frac{\bar{\mu}}{N_i(1 - \bar{\mu})}\), and one where we use the variance stabilizing transformation proposed in the seminal paper by Efron et al.
Comparing both approaches, we can see in Table 2 that they significantly improve upon the MLE, and that the variance stabilized estimator seems to perform a bit better. They are actually not so far off from the oracle linear fit, which has access to the entire season averages as responses.

Figure 6.1: NBA 3pt data. Colors represent the number of 3PT attempts throughout the season.

<table>
<thead>
<tr>
<th>Estimator</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLE</td>
<td>4.38</td>
</tr>
<tr>
<td>JS</td>
<td>3.19</td>
</tr>
<tr>
<td>Transformed JS</td>
<td>2.97</td>
</tr>
<tr>
<td>Linear Fit</td>
<td>2.88</td>
</tr>
</tbody>
</table>

Figure 6.2: Comparison of the MSE between estimators

Python code:

```python
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
```
```python
from sklearn.linear_model import LinearRegression
	sns.set_theme()
sns.color_palette("YlOrBr", as_cmap=True)

nba_data = {}
times = ["Full-Season", "Two-Months"]
for timelapse in times:
    nba_data[timelapse] = pd.read_csv(f"NBA2021-3pt-percentage_{timelapse}.csv")
    .set_index("PLAYER")

for timelapse in times:
    nba_data[timelapse] = nba_data[timelapse].reindex(nba_data["Full-Season"])

nba_full_data = pd.concat([nba_data[timelapse] for timelapse in times], axis=
    f"\{col\}-\{timelapse\}" for timelapse in times for col in nba_data[timelapse].columns]
nba_full_data = nba_full_data.dropna().sort_values(by="3P%-Two-Months")
nba_full_data.head()

# Linear Regression

linReg = LinearRegression().fit(
    nba_full_data["3P%-Two-Months"], nba_full_data["3P%-Full-Season"]
)

nba_full_data["LinRegPred-3P%"] = linReg.predict(nba_full_data["3P%-Two-Months"])

# James Stein Estimator

avg_3PM = sum(
    nba_full_data["3PA-Two-Months"]
    * nba_full_data["3P%-Two-Months"]
    / sum(nba_full_data["3PA-Two-Months"])
)

sigma_PM_sq = (avg_3PM / 100 * (1 - avg_3PM / 100) * np.mean(1 / nba_full_data["3PA-Two-Months"])

n = nba_full_data.shape[0]
```

6-6
slope = 1.0 - (n - 3) * sigma_PM_sq / np.sum(nba_full_data["3P%-Two-Months"] / 100 - avg_3PM / 100) ** 2

nba_full_data["JSPred-3P%"] = (nba_full_data["3P%-Two-Months"] - avg_3PM) * slope + avg_3PM

```python
def compute_JS_varstabilized(X, n_attempts):
    Z = np.sqrt(n_attempts) * np.arcsin(2 * X - 1)
    slope = 1.0 - (len(Z) - 3) / np.sum((Z - Z.mean()) ** 2)
    mu_hat = (Z - Z.mean()) * slope + Z.mean()
    return 0.5 * (np.sin(mu_hat / np.sqrt(n_attempts)) + 1.0)
```

nba_full_data["JSPred-VarStabilized-3P%"] = 100 * compute_JS_varstabilized(nba_full_data["3P%-Two-Months"] / 100, nba_full_data["3PA-Two-Months"].mean)

```python
# Scatter Plot
plt.figure(figsize=(12, 6))
sns.scatterplot(
    x="3P%-Two-Months", y="3P%-Full-Season", palette="icefire", hue="3PA-Full-Season", data=nba_full_data,
)

sns.lineplot(
    x="3P%-Two-Months", y="LinRegPred-3P%", data=nba_full_data, label="Linear Fit",
)
sns.lineplot(
    x="3P%-Two-Months", y="JSPred-VarStabilized-3P%", data=nba_full_data, label="JS Variance Stabilized Estimate",
)
sns.lineplot(
    x="3P%-Two-Months", y="JSPred-3P%", data=nba_full_data, label="JS Variance Stabilized Estimate",
)
```

6-7
x="3P%–Two–Months", y="JSPred–3P%", data=nba_full_data, label="JS_Estimate"

plt.xlim((15, 55))
plt.ylim((15, 55))
plt.savefig("hw6–JS.pdf")
plt.show()

# Mean Squared Error
nba_full_data["MLE–3P%"] = nba_full_data["3P%–Two–Months"]

for method in ["LinRegPred", "JSPred", "JSPred–VarStabilized", "MLE"]:  
mse = np.sqrt(np.mean((nba_full_data[f"{method}–3P%"] – nba_full_data["3P%–Full–Season"]

print(f"MSE for {method}: {mse}")